

ISSN 2311-8806

Modern European Researches

Issue 5 (T.1)

2019



Salzburg, Austria

MODERN EUROPEAN RESEARCHES (2019) ISSUE 5 (T.1), 70 P.

Modern European Researches Journal is the peer review journal, which reflects the most outgoing scientific investigations in such fields of knowledge, as pedagogy, education and training, comprehensive study of human, psychology, social problems of medicine and ecology; philosophy, sociology, political science, jurisprudence, economics; language and literature study, study of art, study of culture.

EDITORIAL BOARD

Olga Bermant-Polyakova, PhD, Israel

Tatyana Fedotova, PhD, Professor, Ukraine

Alla Gabidullina, PhD, Professor, Ukraine

Pavel Gorev, PhD, Associate Professor, Russia

Mariya Greb, PhD, Associate Professor, Ukraine

Natalya Korableva, PhD, Associate Professor, Ukraine

Nikolay Kotryahov, PhD, Professor, Russia

Kanat Lakbaev, PhD, Associate Professor, Kazakhstan

Galina Nekrasova, PhD, Professor, Russia

Aleksander Nosov, PhD, Professor, Russia

Gennadiy Senkevich, PhD, Associate Professor, Ukraine

Samvel Sukiasyan, PhD, Professor, Armenia

Eugene Vechtomov, PhD, Professor, Russia

Elena Visotskaya, PhD, Professor, Ukraine

EDITORIAL ADDRESS

SEEBURGSTRASSE 7,
5201 SEEKIRCHEN AM WALLERSEE,
SALZBURG, AUSTRIA
PUBLISHER@DOAJ.NET

ISSN2311-8806

Authors are responsible for accuracy of the information, contained in the articles.

Editorial opinion can differ from opinion of authors.

If reprinted, the reference to the journal is required.

© All Rights Reserved

Printed in Austria, 2019



CONTENTS

МЕТОДЫ ПОСТРОЕНИЯ ГРАФИКОВ ФУНКЦИЙ, ЗАДАННЫХ В ВИДЕ ПРОИЗВЕДЕНИЯ НЕКОТОРЫХ КВАДРАТИЧНЫХ ФУНКЦИЙ, СОДЕРЖАЩИХ ЗНАК МОДУЛЯ

Ахметова Фания Харисовна, Головина Анастасия Михайловна

4-11

ОБОБЩЕНИЕ ПОНЯТИЯ ПРЕДЕЛА В ВЫСШЕЙ ШКОЛЕ

Безверхний Николай Владимирович

12-17

О МЕТОДИКЕ ИЗЛОЖЕНИЯ НЕКОТОРЫХ РАЗДЕЛОВ ТЕОРИИ АЛГОРИТМОВ: ПРОБЛЕМА ПРИМЕНИМОСТИ ДЛЯ НОРМАЛЬНЫХ АЛГОРИТМОВ МАРКОВА

Белоусов Алексей Иванович

17-36

МЕТОД ЗОЛОТОГО СЕЧЕНИЯ И ДРЕВНЕРУССКИЕ САЖЕНИ

Вергазова Ольга Бухтияровна

36-42

К ЗАДАЧЕ О ФОРМЕ ЯЧЕЕК СОЛНЕЧНЫХ БАТАРЕЙ

Вергазова Ольга Бухтияровна, Кваша Владимир Сергеевич,
Федоров Никита Николаевич, Бекеров Ильмир Дамирович

43-48

ЭЛЕМЕНТАРНОЕ ВВЕДЕНИЕ В ТЕОРИЮ ОДНОМЕРНЫХ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Грибов Александр Федорович, Краснов Игорь Константинович

49-56

МЕТОДИЧЕСКИЕ АСПЕКТЫ ИЗЛОЖЕНИЯ ТЕМЫ «ИНТЕГРИРОВАНИЕ ТРИГОНОМЕТРИЧЕСКИХ ФУНКЦИЙ» В КУРСЕ «НЕОПРЕДЕЛЕННЫЙ ИНТЕГРАЛ»

Ефремова Светлана Николаевна, Косова Анна Владимировна,
Ласковая Татьяна Алексеевна

57-62

ОБ ИНФОРМАТИЗАЦИИ ПРОВЕРКИ ДОМАШНЕГО ЗАДАНИЯ ПО КРИВЫМ И ПОВЕРХНОСТЯМ ВТОРОГО ПОРЯДКА

Кавинов Алексей Владимирович

63-69

МЕТОДЫ ПОСТРОЕНИЯ ГРАФИКОВ ФУНКЦИЙ, ЗАДАННЫХ В ВИДЕ ПРОИЗВЕДЕНИЯ НЕКОТОРЫХ КВАДРАТИЧНЫХ ФУНКЦИЙ, СОДЕРЖАЩИХ ЗНАК МОДУЛЯ

Аннотация

Построение графиков часто облегчает нахождение решений сложных задач, а иногда является единственным средством их решения. Поэтому главное внимание в статье уделяется методам построения графиков, а не изучению отдельных видов функций. Предложены способы построения графиков квадратичных функций, содержащих знак модуля и продемонстрированы практические приемы на примерах различного уровня сложности. Приведены краткие теоретические сведения в области функциональной зависимости таких функций. Цель работы - алгоритмизация построения графиков однозначных и многозначных квадратичных функций, содержащих знак модуля; систематизация методов и приемов, выходящих за пределы школьного учебника математики. Содержание статьи будет полезно старшеклассникам, студентам, а также преподавателям первого курса.

Ключевые слова

модуль, график квадратичной функции, однозначная и многозначная квадратичная функция, криволинейные границы

АВТОРЫ

Ахметова Фания Харисовна,
кандидат физико-математических наук, доцент
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
dobrich2@mail.ru

Головина Анастасия Михайловна,
кандидат физико-математических наук, доцент
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
nastya_gm@mail.ru

Во многих случаях, особенно при решении комплексных задач, наилучшим способом является графическое решение. Главным его преимуществом является наглядность и простота восприятия представляемой информации.

Ранее в работах [1-6] была изложена методика построения графиков **линейных** однозначных и многозначных функциональных зависимостей, содержащих знак модуля. Были рассмотрены общие методы построения графиков, а также метод разбиения функции на отдельные уравнения, метод разбиения координатной плоскости на несколько областей и методика применения среды MathCAD в учебном процессе на примерах исследования некоторых вопросов поведения функции при наличии точек разрыва.

Перейдем к рассмотрению **квадратичной** функции - это функции многочлена с одной или несколькими переменными, в которых член наивысшей степени имеет вторую степень. Ранее в статье [7] уже изучались такие функции.

Несколько усложним задачу и разберем методы построения функций, заданных в виде произведения некоторых квадратичных функций, содержащих знак модуля.

При построении графиков таких функций в основном используется метод раскрытия знака модуля, который позволяет перейти к различным аналитическим видам одной и той же функции в разных областях изменения аргумента. При этом исходный вид функции преобразуется к более простому, уже не содержащему знака модуля. А затем строится в каждой из получившихся областей график своей (уже преобразованной) функции. Но бывают такие функции, которые можно получить и, не применяя данный метод, а пользуясь только свойствами четности/нечетности, симметричности графиков функции $y = g(|x|)$ и $y = |g(x)|$.

Прежде чем перейдем к изложению материала, дадим известные определения четных/нечетных функций и свойства симметричности их графиков, необходимые для дальнейших рассуждений.

Определение 1. Функция $y = f(x)$, $x \in X$, называется **четной**, если область её определения симметрична относительно нуля и для любого значения аргумента $x \in X$ верно равенство: $f(-x) = f(x)$.

График любой четной функции симметричен относительно оси ординат OY .

Определение 2. Функция $y = f(x)$, $x \in X$, называется **нечётной**, если область её определения симметрична относительно нуля и для любого значения аргумента $x \in X$ верно равенство: $f(-x) = -f(x)$.

График любой нечётной функции симметричен относительно начала координат.

Заметим, что не всякая функция является четной или нечётной. В тех случаях, когда функция не является ни четной, ни нечётной, то её называют функцией **общего вида**. График такой функции не обладает симметрией.

Перейдем к рассмотрению конкретных примеров.

Пример 1: Построить график функции $y = x(|x| - 2)$.

Легко видеть, что данная функция является нечетной. Действительно: $y(-x) = -x(|-x| - 2) = -x(|x| - 2)$.

Следовательно, график этой функции симметричен относительно начала координат. В этом случае достаточно построить график функции $y = x(|x| - 2)$ при $x \geq 0$, а затем отобразить его симметрично относительно начала координат. При $x \geq 0$ функция $y = x(|x| - 2)$ примет вид $y = x(x - 2) = x^2 - 2x + 1 - 1 = (x - 1)^2 - 1$.

Таким образом, графиком данной функции является парабола с вершиной в точке $O_1(1, -1)$. Эту точку принимаем за начало новой системы координат $X_1O_1Y_1$. И в новой системе координат строим график функции $y_1 = x_1^2$. Теперь отразим график построенной функции симметрично относительно начала координат.

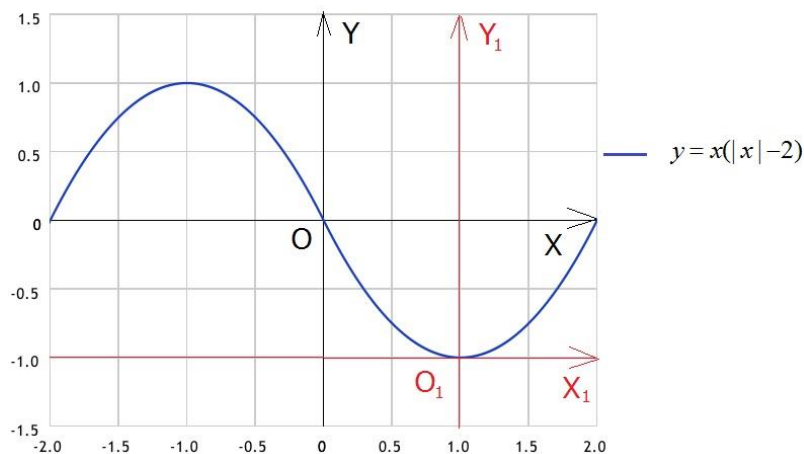


Рис. 1. График функции $y = x(|x| - 2)$

Заметим, что если бы мы не воспользовались свойством симметричности нечетной функции, а раскрывали знак модуля, то получили бы абсолютно тот же график. А именно такой:

$$y = x(|x| - 2) = \begin{cases} x^2 - 2x, x \geq 0, \\ -x^2 - 2x, x < 0. \end{cases}$$

Выделив в каждом из выражений полный квадрат, получим:

$$y = x(|x| - 2) = \begin{cases} (x-1)^2 - 1, x \geq 0, \\ -(x+1)^2 + 1, x < 0. \end{cases}$$

Пример 2: Построить график функции $y = (|x| - 2)(x + 1)$.

Здесь нужно использовать метод раскрытия модуля и перехода к различным аналитическим заданиям данной функции в определенных областях изменения аргумента.

Рассмотрим область $x > 0$, в ней функция примет вид $y = (x - 2)(x + 1) = x^2 - x - 2$, при $x < 0$ функция примет вид $y = (-x - 2)(x + 1) = x^2 - 3x - 2$, при $x = 0$ функция $y = -2$. В итоге получаем кусочную функцию вида:

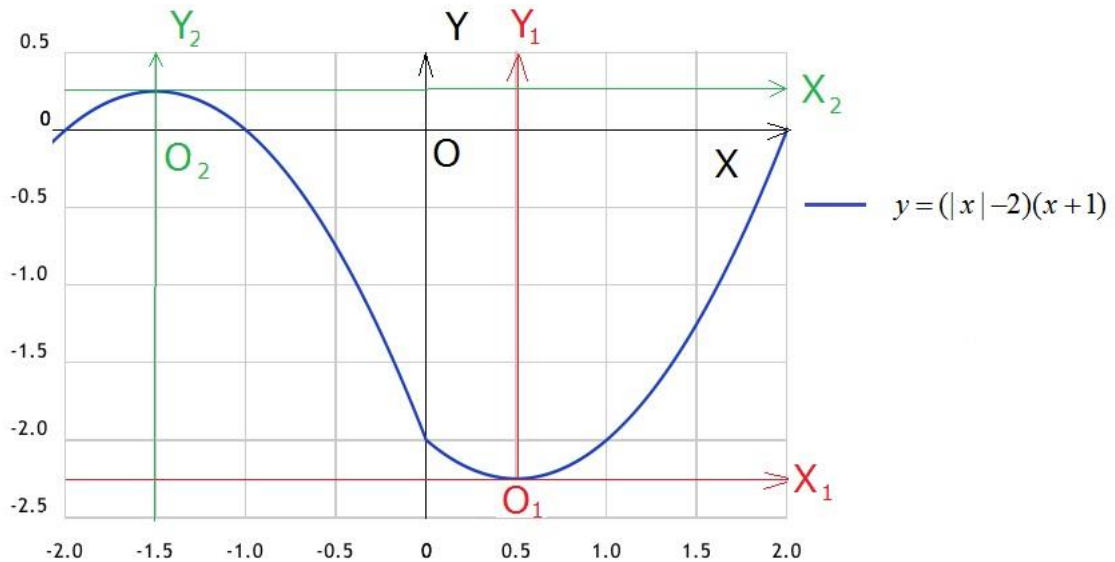
$$y = (|x| - 2)(x + 1) = \begin{cases} x^2 - x - 2, x \geq 0, \\ -x^2 - 3x - 2, x < 0. \end{cases}$$

Строим теперь в каждой из областей $x \geq 0$, $x < 0$ свою функцию. Графиками функций и в первой, и во второй областях являются параболы.

Рассмотрим область $x \geq 0$, а в ней функцию $y = x^2 - x - 2$. Выделим полный квадрат: $y = \left(x - \frac{1}{2}\right)^2 - \frac{9}{4}$. Получаем, что вершина параболы $y = x^2 - x - 2$ находится в точке $O_1\left(\frac{1}{2}, -\frac{9}{4}\right)$. Эту точку принимаем за начало новой системы координат

$X_1O_1Y_1$. Строим в новой системе координат $X_1O_1Y_1$ график функции $y_1 = x_1^2$.

Далее в области $x < 0$ аналогичным образом строим график функции $y = x^2 - 3x - 2$. Выделим полный квадрат: $y = x^2 - 3x - 2 = -\left(x + \frac{3}{2}\right)^2 + \frac{1}{4}$. Таким образом, начало новой системы координат имеет координаты $O_2\left(-\frac{3}{2}, \frac{1}{4}\right)$. В новой системе координат $X_2O_2Y_2$ строим график функции $y_2 = x_2^2$.

Рис. 2. График функции $y = (|x| - 2)(x + 1)$

Рассмотрим теперь примеры построения графика функции вида $y = f(x) \cdot |g(x)|$.

Пример 3: Построить график функции $y = x|x - 2|$.

Используем все тот же метод раскрытия знака модуля, который позволяет перейти к различным аналитическим видам одной и той же функции в разных областях изменения аргумента.

Итак, функция $y = x|x - 2|$ может быть представлена в виде $y = f(x) \cdot |g(x)|$, где $f(x) = x$, $g(x) = x - 2$. Функция $g(x)$ обращается в нуль в точке $x = 2$.

Таким образом, вся числовая ось изменения независимой переменной x разбивается на две области $x < 2$ и $x > 2$. При $x > 2$ функция $y = x|x - 2|$ примет вид: $y = x^2 - 2x$. При $x < 2$ функция $y = x|x - 2|$ имеет вид: $y = -x^2 + 2x$. Строим график функции в каждой из областей.

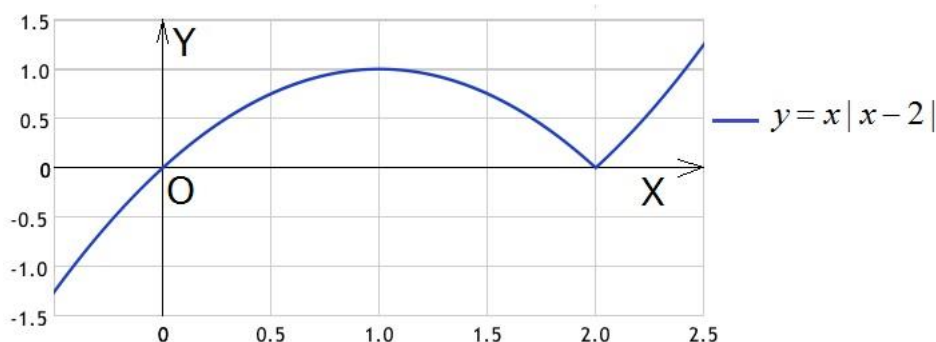
При $x > 2$ функция $y = x|x - 2|$ примет вид: $y = (x - 1)^2 - 1$. Следовательно, вершина параболы - это точка $O_1(1, -1)$.

При $x < 2$ функция примет вид: $y = -(x - 1)^2 + 1$. Следовательно, вершина параболы - это точка $O_2(1, 1)$.

Графиками данных функций являются параболы с вершинами в точках O_1 и O_2 соответственно.

Для функции $y = (x - 1)^2 - 1$ в области $x > 2$ вершиной параболы является точка $O_1(1, -1)$, не принадлежащая самой области. Для функции $y = -(x - 1)^2 + 1$ в области $x < 2$ вершиной параболы является точка $O_2(1, 1)$.

При $x = 2, y = 0$. В каждой из областей $x > 2$ и $x < 2$ строим соответствующую параболу. В итоге получаем график функции $y = x|x - 2| = \begin{cases} (x - 1)^2 - 1, & x \geq 2, \\ -(x - 1)^2 + 1, & x < 2. \end{cases}$

Рис. 3. График функции $y = x|x - 2|$

Построение графика многозначной квадратичной функции

Во всех примерах, разобранных выше, рассматривались **однозначные** квадратичные функции, т.е. каждому значению аргумента соответствовало единственное значение функции. Отметим, что обобщением понятия однозначной функции является **многозначная** функция, которая допускает наличие нескольких значений функции для одного аргумента.

График многозначной квадратичной функции $|y| = ax^2 + bx + c$ строится аналогично графику линейной функции $|y| = ax + b$.

Алгоритм построения графика многозначной квадратичной функции:

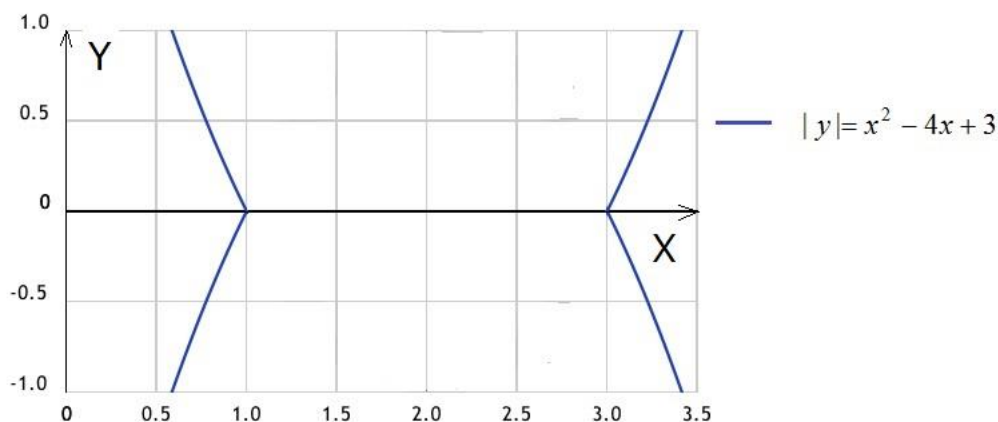
$$|y| = ax^2 + bx + c$$

1. Строим график функции $y = ax^2 + bx + c$ при $y \geq 0$.

2. Отражаем построенный график функции симметрично относительно оси абсцисс.

Пример 4: Построить ГМТ (геометрическое место точек), удовлетворяющих уравнению $|y| = x^2 - 4x + 3$.

При $y \geq 0$ получаем функции $y = x^2 - 4x + 3$. Выделим полный квадрат: $y = x^2 - 4x + 3 = (x - 2)^2 - 1$. Строим график этой функции. Заметим, что нулями функции являются точки $x = 3, x = 1$. Зеркально отражаем построенный график относительно оси абсцисс.

Рис. 4. График функции $|y| = x^2 - 4x + 3$

Несколько усложним задачу, рассмотрим модуль и левой, и правой частей уравнения.

Пример 5: Построить ГМТ, удовлетворяющих уравнению $|y| = |x^2 - 4x + 3|$.

1. Строим график функции $y = |x^2 - 4x + 3|$. Для построения данного графика построим график функции $y = x^2 - 4x + 3$. Нижнюю часть этого графика отразим симметрично относительно оси абсцисс.

2. Отражаем построенный график функции $y = |x^2 - 4x + 3|$ относительно оси абсцисс Ox .

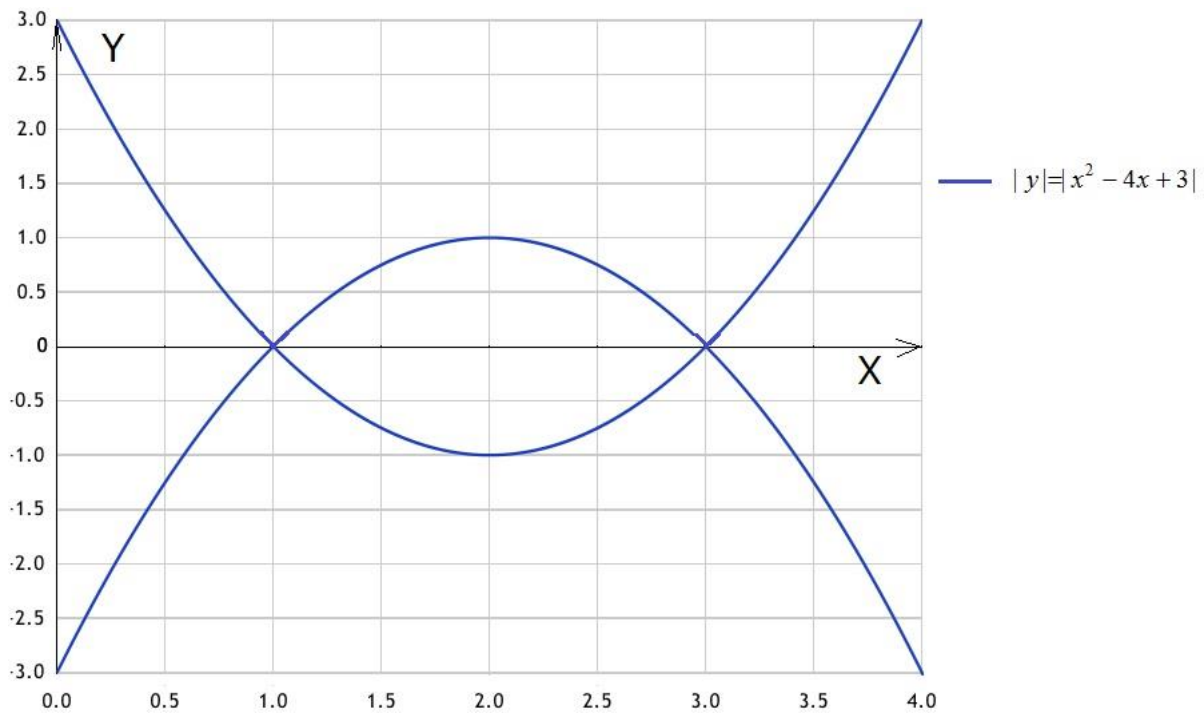


Рис. 5. График функции $|y| = |x^2 - 4x + 3|$

Случай криволинейных границ

Плоскость XOY может разбиваться на две или несколько частей не только любой прямой или прямыми, но и любой кривой, например, параболой. Уравнения границ данных кривых определяются приравниванием к нулю выражения, стоящего под знаком модуля.

Алгоритм построения графика функции $|y - f(x)| = g(x)$

1) Разбиваем плоскость XOY на несколько частей, строя график функции $y = f(x)$, т.е. получаем криволинейные границы.

2) Раскрываем модуль в каждой из этих плоскостей с нужным знаком и находим аналитическое представление функции $|y - f(x)| = g(x)$ в каждой из полученных областей.

3) Строим в каждой из этих областей графики соответствующих функций.

Пример 6: Построить ГМТ плоскости, удовлетворяющих уравнению $|y - x^2 + 2x + 1| - x = 2$. В этом случае, линией, разбивающей плоскость XOY будет парабола $y = x^2 - 2x - 1$. Для построения данной параболы выделим полный квадрат: $y = x^2 - 2x - 1 = (x - 1)^2 - 2$.

При $y \geq x^2 - 2x - 1$ исходное уравнение примет вид: $y - x^2 + 2x + 1 - x - 2 = 0$
или $y = \left(x - \frac{1}{2}\right)^2 - \frac{5}{4}$.

Таким образом, ГМТ, удовлетворяющих исходному уравнению, будет являться кривая $y = \left(x - \frac{1}{2}\right)^2 - \frac{5}{4}$, заключенная между ветвями параболы $y = (x - 1)^2 - 2$.

При $y < x^2 - 2x - 1$ исходное уравнение примет вид:
 $-y + x^2 - 2x - 1 - x - 2 = 0$ или $y = \left(x - \frac{3}{2}\right)^2 - \frac{21}{4}$.

А вне ветвей параболы $y = (x - 1)^2 - 2$ ГМТ исходного уравнения будет являться параболой $y = \left(x - \frac{3}{2}\right)^2 - \frac{21}{4}$.

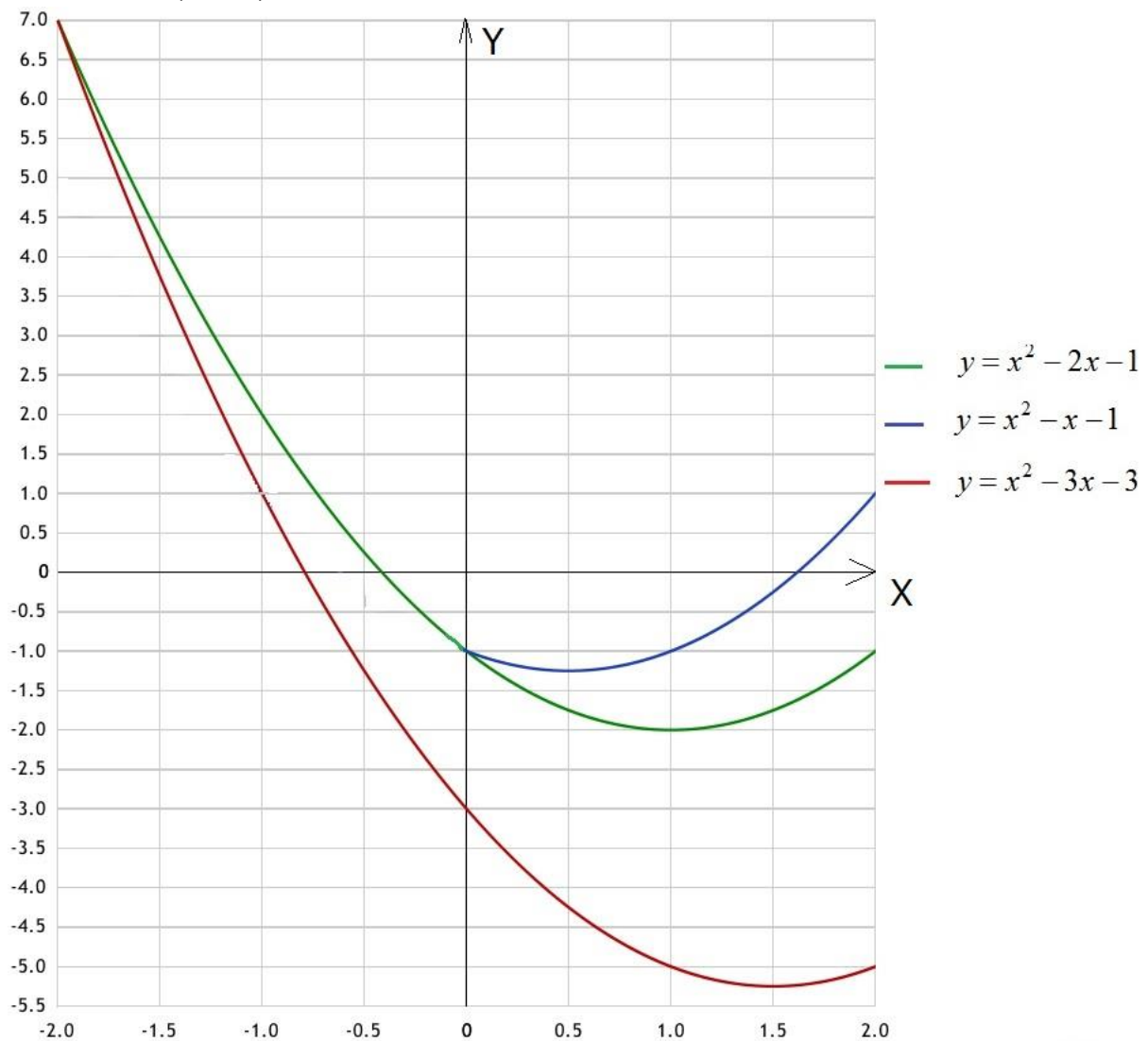


Рис. 6. График функции $|y - x^2 + 2x + 1| - x = 2$

На примерах задач были проиллюстрированы алгоритмы построения графиков однозначных и многозначных функций, заданных в виде произведения некоторых

квадратичных функций, содержащих знак модуля. Описаны все тонкости при выполнении заданий такого рода.

При этом отметим важность обучения студентов различным методам решения задач и построения графиков. В ряде случаев графики существенно облегчают нахождение решений уравнений и неравенств, сокращая и упрощая аналитические выкладки. Необходимо, чтобы каждый студент был готов к выбору наиболее целесообразного и эффективного пути рассуждения. Большое внимание следует уделять накоплению у студентов опыта самостоятельного поиска решений поставленной задачи. Активное использование техники построения графиков различного уровня сложности позволит сформировать навыки и профессиональные компетенции у студентов технических специальностей.

ССЫЛКИ НА ИСТОЧНИКИ

1. Ахметова Ф.Х., Головина А.М. Методика построения графиков линейных функций, содержащих знак модуля // Научно-методический электронный журнал «Концепт». – 2017. – № 5 (май). – С. 159–170. – URL: <http://e-koncept.ru/2017/170117.htm>.
2. Ахметова Ф.Х., Головина А.М. Линейная комбинация функций, содержащих знак модуля и методика построения их графиков // Научно-методический электронный журнал «Концепт». – 2017. – № V6. – С. 49–54. – URL: <http://e-koncept.ru/2017/170138.htm>.
3. Ахметова Ф. Х., Буякевич А. Е. Исследование некоторых вопросов поведения функций и построение графиков с привлечением среды MathCAD // Научно-методический электронный журнал «Концепт». – 2017. – № V9. – С. 77–87. – URL: <http://e-koncept.ru/2017/171024.htm>.
4. Ахметова Ф. Х., Головина А. М. Метод разбиения функции на отдельные уравнения при построении графиков многозначных линейных функций, содержащих знак модуля // Научно-методический электронный журнал «Концепт». – 2018. – № V9. – С. 76–81. – URL: <http://e-koncept.ru/2018/186090.htm>.
5. Ахметова Ф. Х., Головина А. М. Построение графиков многозначных линейных функций, содержащих знак модуля // Научно-методический электронный журнал «Концепт». – 2018. – № V11. – С. 6–11. – URL: <http://e-koncept.ru/2018/186105.htm>.
6. Ахметова Ф. Х., Головина А. М. Метод разбиения плоскости на несколько областей при построении графиков многозначных линейных функций, содержащих знак модуля // Научно-методический электронный журнал «Концепт». – 2018. – № V12. – С. 1–6. – URL: <http://e-koncept.ru/2018/186118.htm>.
7. Ахметова Ф. Х., Головина А. М. Методы построения графиков квадратичных функций, содержащих знак модуля. Modern European Researches. – 2019. – №2(1). – С. 4–10. <https://elibrary.ru/contents.asp?titleid=53202>

Faniya Akhmetova

Candidate of Physical-Mathematical Sciences, Associate Professor, Bauman Moscow State Technical University, Moscow

dobrich2@mail.ru

Anastasiya Golovina

Candidate of Physical-Mathematical Sciences, Associate Professor, Bauman Moscow State Technical University, Moscow

nastya_gm@mail.ru

Methods for constructing graphs of functions defined as the product of some quadratic functions containing the modulus sign

Abstract. Graphing often makes it easier to find solutions to complex problems, and sometimes it is the only way to solve them. Therefore, the main attention in the article is paid to graphing methods, and not to the study of certain types of functions. Methods are proposed for constructing graphs of quadratic functions containing the sign of the module and practical techniques are demonstrated using examples of various levels of complexity. Brief theoretical information is given in the field of the functional dependence of such functions. The purpose of the work is the graphization algorithm for the unique and multi-valued quadratic functions containing the sign of the module; systematization of methods and techniques that go beyond the school textbook of mathematics. The content of the article will be useful to high school students, students, as well as first-year teachers.

Key words: module, graph of a quadratic function, single-valued and multi-valued quadratic function, curved borders.

ОБОБЩЕНИЕ ПОНЯТИЯ ПРЕДЕЛА В ВЫСШЕЙ ШКОЛЕ

Аннотация

В статье сравниваются различные подходы к понятию предела, подробно рассматривается понятие интеграла Римана как предела интегральных сумм и приводится понятие предела функции по базе. Тем самым, даётся обобщение понятий предела числовой последовательности, функции в точке и на бесконечности и других, а также обосновываются свойства предела функции по базе. Все известные по первым двум курсам университетов понятия предела трактуются как пределы по соответствующим базам, что позволяет доказать известные свойства пределов по базе и применять их во всех известных ситуациях. Наиболее важным является обоснование свойств предела интегральных сумм, не применяя которые, невозможно обосновать такие свойства интеграла Римана, как линейность, интегрирование неравенств и другие. Цель работы - дать теоретическое обоснование для применения свойств интеграла Римана. Содержание статьи полезно студентам и преподавателям университетов.

Ключевые слова

база, предел функции по базе, интеграл Римана, интеграл Лебега

АВТОР

Безверхний Николай Владимирович,
доцент кафедры математического моделирования
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
nbezv@mail.ru

Введение

Всем обладателям диплома о высшем техническом образовании известны определения понятий предела числовой последовательности, функции в точке и на бесконечности, одностороннего предела функции в точке. Эти понятия приводятся в рамках стандартных курсов математического анализа и снабжены теоремами о свойствах введенных понятий предела. Каждый раз, вводя очередное понятие предела, доказывают его свойства: линейность, предельный переход в неравенствах и другие [1].

Оказывается, все перечисленные понятия вместе с теоремами об их свойствах подпадают под одно общее определение предела функции по базе [2]. И доказывать свойства различных пределов можно только один раз - для предела функции по базе. Этим мы займёмся в первой части статьи.

Вторая часть посвящена интегралам Римана и Лебега. Как известно, интеграл Римана определяется как предел интегральных сумм при измельчении разбиения области интегрирования. Но далеко не каждый лектор открывает слушателям глаза на то, что интегральная сумма не является ни одним из тех объектов, о которых говорилось ранее при обсуждении понятия предела: она не является ни числовой последовательностью, ни функцией действительного и даже векторного аргумента. Строго говоря, интегральная сумма представляет собой отображение из множества размеченных разбиений области интегрирования в поле действительных чисел. И этот факт

не позволяет нам безосновательно применять свойства предела функции, утверждая, например, что интеграл Римана обладает свойством линейности.

Определение предела по базе

Определение 1. Семейство S подмножеств B множества X называется базой в множестве X , если выполнены условия:

- 1) $\forall B \in S (B \neq \emptyset)$
- 2) $\forall B_1, B_2 \in S \exists B_3 \in S (B_3 \subset B_1 \cap B_2)$.

Таким образом, элементы семейства S - непустые подмножества множества X , и в пересечении любых двух из них содержится некоторый элемент из этого семейства.

Определение 2. Пусть $f: X \rightarrow R$ - функция, определённая на множестве X со значениями в поле действительных чисел. Пусть S - база в X . Число A называется пределом функции $f: X \rightarrow R$ по базе S , если для любой окрестности $V(A)$ точки A найдётся элемент $B \in S$ базы, образ которого $f(B)$ содержится в окрестности $V(A)$.

Для обозначения того, что число является пределом функции по базе, используют запись

$$\lim_S f(x) = A.$$

Запишем определение предела функции по базе в логической символике:

$$\lim_S f(x) = A \Leftrightarrow \forall V(A) \exists B \in S (f(B) \subset V(A)).$$

Учитывая то, что множество значений функции $f(x)$ является полем действительных чисел, следует иметь в виду и следующую формулировку приведённого определения:

$$\lim_S f(x) = A \Leftrightarrow \forall \varepsilon > 0 \exists B \in S (\forall x \in B \Rightarrow |f(x) - A| < \varepsilon).$$

Определение 3. Функция $f(x)$ называется бесконечно малой по базе S , если

$$\lim_S f(x) = 0.$$

Функция $f(x)$ называется ограниченной по базе S , если

$$\exists B \in S, \exists C > 0: \forall x \in B \Rightarrow |f(x)| \leq C.$$

Свойства предела по базе

Теорема 1. Если существует предел функции $f: X \rightarrow R$ по базе S , то он единственный.

Доказательство. Предположим, что числа $A < B$ удовлетворяют определению предела функции f по базе S . Рассмотрим непересекающиеся окрестности $V(A), V(B)$. По определению предела по базе существуют элементы базы $B_1, B_2 \in S$, для которых справедливы утверждения: $\forall x \in B_1 \Rightarrow f(x) \in V(A)$ и $\forall x \in B_2 \Rightarrow f(x) \in V(B)$. Существует элемент базы $B_3 \subset B_1 \cap B_2$. При этом $\forall x \in B_3 \Rightarrow f(x) \in V(A) \cap V(B) = \emptyset$. Итак, поскольку выбранные нами окрестности точек A и B не пересекаются, то получено противоречие, позволяющее сделать вывод об ошибочности предположения $A < B$. Теорема доказана.

Теорема 2. Если существуют пределы функций $f_1 : X \rightarrow R$ $f_2 : X \rightarrow R$ по базе S , и они равны A_1, A_2 , соответственно, то существует предел суммы $f_1 + f_2 : X \rightarrow R$, и он равен $A_1 + A_2$.

Доказательство.

Пусть $\lim_S f_1(x) = A_1, \lim_S f_2(x) = A_2 \Leftrightarrow \forall \varepsilon > 0 \exists B_1 \in S (\forall x \in B_1 \Rightarrow |f_1(x) - A_1| < \varepsilon/2),$

$\exists B_2 \in S (\forall x \in B_2 \Rightarrow |f_2(x) - A_2| < \varepsilon/2)$ Поскольку $\exists B_3 \subset B_1 \cap B_2$, то $(\forall x \in B_3 \Rightarrow |f_1(x) - A_1| < \varepsilon/2, |f_2(x) - A_2| < \varepsilon/2) \Rightarrow |(f_1(x) + f_2(x)) - (A_1 + A_2)| < \varepsilon$.

Теорема 3. Если существуют пределы функций $f_1 : X \rightarrow R$ $f_2 : X \rightarrow R$ по базе S , и они равны A_1, A_2 , соответственно, то существует предел произведения функций $f_1 f_2 : X \rightarrow R$, и он равен $A_1 A_2$.

Доказательство.

Пусть $\lim_S f_1(x) = A_1, \lim_S f_2(x) = A_2 \Leftrightarrow \forall \varepsilon > 0 \exists B_1 \in S (\forall x \in B_1 \Rightarrow |f_1(x) - A_1| < \varepsilon),$

$\exists B_2 \in S (\forall x \in B_2 \Rightarrow |f_2(x) - A_2| < \varepsilon)$ Поскольку $\exists B_3 \subset B_1 \cap B_2$, то

$(\forall x \in B_3 \Rightarrow |f_1(x) - A_1| < \varepsilon, |f_2(x) - A_2| < \varepsilon) \Rightarrow$

$|(f_1(x)f_2(x)) - (A_1A_2)| = |(f_1(x)f_2(x) - f_1(x)A_2) + (f_1(x)A_2 - A_1A_2)| \leq |f_1(x)||f_2(x) - A_2| + |f_1(x) - A_1||A_2| < |f_1(x)|\varepsilon + |A_2|\varepsilon$. Из существования предела функции $f_1(x)$ по базе следует её ограниченность по базе: $\exists B_4 \in S, \exists C > 0 : \forall x \in B_4 \Rightarrow |f_1(x)| \leq C$, поэтому

$\forall x \in B_4 \cap B_3 \Rightarrow |f_1(x)| \leq C$ и в итоге

$|(f_1(x)f_2(x)) - (A_1A_2)| < C\varepsilon + |A_2|\varepsilon$. Из произвольности $\varepsilon > 0$ следует справедливость утверждения теоремы 3.

Замечание. Для частного и разности функций верны соответствующие утверждения о пределах. Мы не будем их доказывать.

Теорема 4. Если существуют пределы A_1, A_2 функций $f_1 : X \rightarrow R$ $f_2 : X \rightarrow R$ по базе S , и при всех значениях аргумента $x \in X$ выполняется неравенство $f_1(x) < f_2(x)$, то пределы удовлетворяют неравенству $A_1 \leq A_2$.

Доказательство. Как и выше, предполагаем существование пределов: $\lim_S f_1(x) = A_1, \lim_S f_2(x) = A_2$. Если $A_1 > A_2$, то можно рассмотреть окрестности

$V(A_1), V(A_2) : V(A_1) \cap V(A_2) = \emptyset$. Пусть число $0 < \varepsilon < \frac{A_1 - A_2}{2}$. Из существования пределов

следует: $\exists B_1 \in S (\forall x \in B_1 \Rightarrow |f_1(x) - A_1| < \varepsilon)$ $\exists B_2 \in S (\forall x \in B_2 \Rightarrow |f_2(x) - A_2| < \varepsilon)$. Но тогда

$\exists B_3 \in S : B_3 \subset B_1 \cap B_2, (\forall x \in B_3 \Rightarrow |f_1(x) - A_1| < \varepsilon, |f_2(x) - A_2| < \varepsilon)$. Это означает, что для всех $x \in B_3 \Rightarrow f_1(x) > f_2(x)$, что противоречит условию теоремы. Этим противоречием заканчивается доказательство.

Примеры баз в различных разделах математического анализа

Начнём с теории числовых последовательностей. Как известно, число A называется пределом числовой последовательности $\{x_n\}$, если для любого положительного числа ε найдётся натуральный номер $N(\varepsilon)$, для которого выполнено следующее:

$$\forall n > N(\varepsilon) \Rightarrow |x_n - A| < \varepsilon.$$

Удобно трактовать числовую последовательность как функцию натурального аргумента с действительными значениями. Таким образом, в случае числовых последовательностей $X = \{1, 2, \dots, n, \dots\}$, $f(n) = x_n$, $S = \{(n, n+1, \dots), n \in N\}$. Поясним последнее: система S подмножеств натурального ряда N - это все «лучи» из натуральных чисел вида $(n, n+1, \dots)$.

Проверим выполнение требований, предъявляемых к элементам базы. Во-первых, любой «луч» не является пустым множеством, поскольку содержит все натуральные числа, большие или равные n . Во-вторых, любые два «луча»

$(n, n+1, \dots)$, $(m, m+1, \dots)$, $n \leq m$ имеют непустое пересечение $(m, m+1, \dots)$, которое является элементом базы.

Таким образом, ссылаясь на теоремы о свойствах предела функции по базе, можно пользоваться свойствами предела числовой последовательности, не доказывая их.

Обсудим понятие предела функции в точке a : число A называется пределом функции $f(x)$, если для любого положительного числа ε найдётся положительное $\delta(\varepsilon)$, для которого выполнено следующее: $\forall x: 0 < |x - a| < \delta(\varepsilon) \Rightarrow |f(x) - A| < \varepsilon$. Будем считать, что наша функция определена на интервале (c, d) , содержащем точку a . Тогда в качестве базы можно использовать множество всех проколотых интервалов вида $(a - \varepsilon, a + \varepsilon) \subset (c, d)$ с выброшенной точкой a .

Ясно, что все они не пусты, и любые два проколотых интервала с центром a $(a - \varepsilon_1, a + \varepsilon_1) \subset (c, d)$, $(a - \varepsilon_2, a + \varepsilon_2) \subset (c, d)$, $\varepsilon_1 \leq \varepsilon_2$ содержат в своём пересечении проколотый интервал меньшей длины: $(a - \varepsilon_1, a + \varepsilon_1) \subset (a - \varepsilon_1, a + \varepsilon_1) \cap (a - \varepsilon_2, a + \varepsilon_2)$, который тоже принадлежит рассматриваемой базе.

Таким образом, ссылаясь на теоремы о свойствах предела функции $f(x)$ по базе, можно пользоваться свойствами предела функции в точке, не доказывая их.

Далее рассмотрим базу, соответствующую пределу функции $f(x)$ на бесконечности. Она может быть определена так: $S = \{|x| > d, x \in R, d > D > 0\}$ - это множество всех «двусторонних лучей» с вершинами $\pm d$, где все $d > D > 0$. Ясно, что все эти множества не пусты, и в пересечении любых двух содержится третье: при $d_1 \geq d_2$ $[(-\infty, d_1) \cup (d_1, +\infty)] \subset [(-\infty, d_1) \cup (d_1, +\infty)] \cap [(-\infty, d_2) \cup (d_2, +\infty)]$.

Таким образом, ссылаясь на теоремы о свойствах предела функции $f(x)$ по базе, можно пользоваться свойствами предела функции на бесконечности, не доказывая их.

Аналогично строятся базы для пределов функции при стремлении аргумента к $+\infty$ и к $-\infty$: в первом случае $S = \{(d, +\infty), d > D\}$, во втором $S = \{(-\infty, d), d < D\}$.

При рассмотрении односторонних пределов функции в точке используют базы из полуинтервалов. Для левого одностороннего предела - полуинтервалы $S = \{(a - \varepsilon, a), \varepsilon > 0\}$, а для правого - полуинтервалы $S = \{(a, a + \varepsilon), \varepsilon > 0\}$.

Интегралы Римана и Лебега как пределы интегральных сумм по базе

Начнём с интеграла Римана. Рассмотрим функцию $f(x)$, определённую на отрезке $[a, b]$. Обозначим разбиение отрезка $[a, b]$ точками $x_0 = a < x_1 < \dots < x_n = b$ через $T_{[a, b]} = T$. На каждом отрезке $[x_k, x_{k+1}]$ возьмём произвольную точку $z_k \in [x_k, x_{k+1}]$.

Набор Z точек $z_1 < \dots < z_n$ называется разметкой, а разбиение $T_{[a,b]} = T$ с разметкой Z называется размеченным разбиением и обозначается T_Z .

Определим основной в рамках этой статьи объект исследования: интегральной суммой функции $f(x)$ на размеченном разбиении T_Z называется отображение множества всех размеченных разбиений отрезка $[a, b]$ в множество действительных чисел, заданное формулой: $S_f(T_Z) = \sum_{k=1}^n f(z_k)(x_{k+1} - x_k)$. Так определённое отображение не является ни числовой последовательностью, ни числовой функцией действительного аргумента, ни функцией векторного аргумента. Поэтому для определения интеграла Римана как предела интегральных сумм воспользуемся понятием предела функции по базе.

Рассмотрим базу на множестве всех размеченных разбиений отрезка $[a, b]$: обозначим через B_δ множество всех размеченных разбиений отрезка $[a, b]$ с диаметром, меньшим δ . Напомним, что диаметр d разбиения (размеченного разбиения) отрезка $[a, b]$ - это максимум длин отрезков $[x_0, x_1], \dots, [x_{n-1}, x_n]$. Тогда все множества B_δ при положительных δ образуют базу на множестве всех размеченных разбиений. Свойства базы легко проверяются.

Определение 4. Определённым интегралом (интегралом Римана) функции $f(x)$ по отрезку $[a, b]$ называется предел интегральных сумм по выбранной выше базе

S в множестве размеченных разбиений: $\lim_S S_f(T_Z) = \int_a^b f(x)dx$.

Ссылаясь на свойства предела по базе, приходим к таким свойствам интеграла Римана, как линейность, интегрирование неравенств, оценка по модулю.

Далее рассмотрим интеграл Лебега [3] от измеримой функции как предел интегральных сумм по соответствующей базе. Будем считать, что функция $f(x)$ измерима, и все её значения принадлежат отрезку $[A, B]$, расположенному на оси Oy .

Пусть $T: A = y_0 < y_1 < \dots < y_n = B$ - разбиение отрезка $[A, B]$. Определим множества $e_k = f^{-1}[y_k, y_{k+1})$ как прообразы промежутков $[y_k, y_{k+1})$. Тогда верхней суммой Лебега называют отображение множества разбиений отрезка $[A, B]$ в множество действительных чисел, заданное формулой $S = \sum_{k=0}^{n-1} y_{k+1} me_k$, где me_k - мера Лебега множества e_k .

Аналогично определяются нижние суммы: $S = \sum_{k=0}^{n-1} y_k me_k$. Можно доказать, что при измельчении разбиения $T: A = y_0 < y_1 < \dots < y_n = B$, то есть при добавлении новых точек $y_i < y'_i < y_{i+1}$ нижние суммы не уменьшаются, а верхние не увеличиваются. При этом все нижние суммы остаются меньше верхних сумм. Это означает существование точных граней U и V у этих множеств.

Используя принятое выше обозначение d для диаметра разбиения $T: A = y_0 < y_1 < \dots < y_n = B$, можно доказать, что $0 \leq S - s \leq dm(E)$, где E - область определения измеримой функции $f(x)$ - измеримое множество. И далее: из произвольности d , вытекающей из возможности неограниченно измельчать разбиение

$T : A = y_0 < y_1 < \dots < y_n = B$, следует равенство пределов, совпадающих, благодаря монотонности сумм Лебега, с точными гранями U, V , то есть $U=V$.

Такие рассуждения позволяют определить интеграл Лебега от измеримой функции как общее значение точных граней верхних и нижних сумм $U=V$. Теперь становится ясно, что, как и в случае интеграла Римана, интеграл Лебега представляет собой предел интегральных сумм по базе на множестве всех, например, верхних сумм Лебега. База задаётся аналогично рассмотренной для интеграла Римана. Дальнейшее продвижение в этом направлении упирается в непреодолимое препятствие: рассматривая сумму двух измеримых функций, мы для каждой из них определяем свой отрезок $[A, B]$. Поэтому не удаётся связать интегралы Лебега от двух слагаемых с предельным переходом по общей базе на одном и том же множестве.

Кроме того, существует и другой подход к определению интеграла Лебега - через простые функции, и он позволяет обойтись только понятием предела числовой последовательности [4].

Заключение

При изложении таких разделов математического анализа, как теория последовательностей, пределов функций, интеграла Римана имеет смысл обобщать понятия всех перечисленных пределов понятием предела по базе, что в случае интеграла Римана просто необходимо для строгого обоснования всех его свойств.

С интегралом Лебега такой подход желателен только на стадии определения, использующего понятие сумм Лебега. Для дальнейшего изложения теории он оказывается бесполезен. Та же ситуация складывается и при определении интеграла Лебега через простые функции.

ССЫЛКИ НА ИСТОЧНИКИ

1. Ильин В.А., Позняк Э.Г. Основы математического анализа. В 2-х ч. Ч. 1-я Физматлит 2005, Ч. 2-я 2002.
2. Зорич В.А. Математический анализ. Ч. 1. Фазис 1997.
3. Натансон И.П.. Теория функций вещественной переменной. Лань 2008.
4. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа. Наука 1976.

О МЕТОДИКЕ ИЗЛОЖЕНИЯ НЕКОТОРЫХ РАЗДЕЛОВ ТЕОРИИ АЛГОРИТМОВ: ПРОБЛЕМА ПРИМЕНИМОСТИ ДЛЯ НОРМАЛЬНЫХ АЛГОРИФМОВ МАРКОВА

Аннотация

Актуальность рассматриваемой методической задачи обусловлена тем, что в учебных планах студентов-программистов существенное место занимает теория алгоритмов, и требуется отработка методики строгого и в то же время доступного студентам, не специализирующимся собственно в математике, изложения основ этой теории. В статье предлагается методика изложения ключевых положений теории алгоритмов на примере нормальных алгорифмов Маркова. Подробно доказывается алгоритмическая неразрешимость проблемы применимости (или проблемы останова) для нормальных алгорифмов. Кратко излагаются основные результаты, необходимые для понимания основной теоремы. Содержание статьи будет полезно студентам программистских

специальностей, а также преподавателям курсов математической логики и теории алгоритмов, курсов проектирования алгоритмов.

Ключевые слова

алгоритм, нормальный алгорифм Маркова,
алгоритмически неразрешимые проблемы

АВТОР

Белоусов Алексей Иванович,
кандидат физико-математических наук, доцент
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
al_belous@bk.ru

Введение

В учебные планы по специальностям, связанным так или иначе с программными технологиями, обязательно входят курсы, посвященные разработке различных алгоритмов. Обстоятельный обзор таких алгоритмов содержится в недавно вышедшей в русском переводе книги [1].

Это, несмотря на возникающие собственно математические вопросы, - практические аспекты теории алгоритмов как таковой. Фундаментом же является классическая теория алгоритмов, представленная многочисленными точными моделями алгоритма (машины Тьюринга, рекурсивные функции, канонические системы Поста, нормальные алгорифмы Маркова). Эта теория должна лежать в основе всех дальнейших разработок, и главный результат здесь - анализ алгоритмических массовых проблем и доказательство неразрешимости некоторых из них. Любой программист должен четко знать границы применения алгоритмических методов, понимать суть алгоритмической неразрешимости, знать, какие алгоритмически неразрешимые проблемы возникают в сфере программирования (например, проблема функциональной эквивалентности программ [2, 3]).

В данной статье рассматривается методика изложения ключевых положений теории алгоритмов на примере нормальных алгорифмов Маркова. Эта модель алгоритма заслуживает особого внимания, так как лежит в основе некоторых языков программирования (например, языка РЕФАЛ [4, 5, 6]), предназначенных для обработки символьной информации, для разработки трансляторов в том числе. Изложение ведется на доступном для понимания студентов-нематематиков (программистов, например) уровне с опорой на основополагающие источники, книгу [7] прежде всего. Следует иметь в виду и очень хорошую книгу Б.А. Кушнера [8]. Мы стараемся излагать все доказательства максимально просто и прозрачно, не нанося существенного ущерба строгости изложения. Собственно, в этом и состоит суть предлагаемой методики: достаточно просто и в то же время строго изложить часто весьма непростые результаты и доказательства. Методика ориентирована на студентов-нематематиков, но тем, которым в своей профессиональной деятельности придется математику использовать.

Предваряя дальнейшее изложение основных положений теории алгоритмов, необходимо дать перечень главных разделов этой теории. Эти разделы таковы:

1. Определение основной модели (в данном случае, нормального алгорифма) и формулировка основной гипотезы, состоящей в том, что предлагаемая модель представляет, в известном смысле, любой алгоритм.

2. Понятие эквивалентных алгоритмов.

3. Теоремы сочетания (основные конструкции из алгоритмов, как то: последовательное применение, независимое применение, условный переход и цикл), согласно которым любое сочетание алгоритмов может быть в конечном счете сведено к основной модели.

4. Универсальный алгоритм (по сути, универсальный интерпретатор в классе программ, определяемых основной моделью).

5. Разрешимые и перечислимые множества.

6. Алгоритмически неразрешимые проблемы.

Материал статьи основан на курсе «Логика и теория алгоритмов», который автор уже многие годы читает на программистских специальностях МГТУ им. Н.Э. Баумана. Студенты подготовлены к освоению курса курсом дискретной математики, опирающимся на базовый учебник [9], в котором для понимания теории алгоритмов особенно важны элементы теории формальных языков.

Цель статьи состоит прежде всего в том, чтобы очертить рамки предлагаемой методики. Поэтому мы ограничиваемся минимумом примеров. Подробному обсуждению примеров и приемов разработки нормальных алгоритмов для решения конкретных задач будет посвящена отдельная статья.

Концепция алгоритма

Прежде чем определять алгоритм математически точно, нужно кратко охарактеризовать саму концепцию алгоритма, или то, что называют **алгоритмом в интуитивном смысле слова**. Здесь методически целесообразно ввести основные обозначения и понятия, используемые в дальнейшем.

Алгоритм можно изначально рассматривать как «черный ящик», осуществляющий преобразование входных данных из некоторого множества X в результаты, составляющие множество Y . Запишем условно это так

$$A: X \longrightarrow Y \quad (1)$$

и будем говорить: **алгоритм A типа (X, Y)** .

Входные данные и результаты алгоритма понимаются как **конструктивные объекты**. Очень коротко: конструктивный объект есть (неформально) некий объект, который может быть построен за конечное число шагов по определенным правилам («правилам сборки») из неких базовых элементов (наподобие моделей в детском конструкторе).

Всюду в дальнейшем мы будем рассматривать частный случай конструктивного объекта, а именно, **слово в конечном алфавите** [9]. Здесь базовые элементы - буквы алфавита, а единственное правило сборки - известная операция соединения слов. Это частный случай, но важнейший. Можно выдвинуть гипотезу, согласно которой любой конструктивный объект может быть представлен (более или менее сложно) в виде слова в конечном алфавите.

Итак, мы можем уточнить идею алгоритма, понимаемого как преобразователь слов в конечных алфавитах.

Не всякий преобразователь слов, однако, будет алгоритмом. Алгоритм характеризуется следующими тремя признаками:

- **признак детерминированности**: алгоритм есть предписание, определяющее пошаговый процесс преобразования входных слов, в котором каждый шаг определен однозначно, а также однозначно определены условия прекращения процесса;

- **признак массовости**: алгоритм есть такое предписание, которое единообразно применяется в широком классе входных данных независимо от их индивидуальных особенностей (например, алгоритм вычисления корней квадратного уравнения

един для всех уравнений и никак - сам по себе - не зависит от коэффициентов конкретного уравнения);

- **признак результативности:** алгоритм есть такое предписание, в силу которого процесс преобразования входных данных (слов) завершается для достаточно широкого множества этих данных выдачей результата (также в виде слова в некотором конечном алфавите).

Из перечисленных признаков первый есть важнейший. Процесс работы алгоритма не допускает никакого выбора шага. Здесь методически уместно провести сравнение алгоритмического процесса преобразования входных данных с игровым. Например, шахматная партия тоже есть некий процесс преобразования конструктивных объектов (позиций на доске, известным образом представляемых вербально), но на каждом шаге возможен выбор в рамках правил игры, то есть очередной шаг (ход в игре) не определен однозначно. В процессе работы алгоритма такая ситуация невозможна. В курсе математической логики, читаемом вслед за теорией алгоритмов, студентам объясняется, что подобный же игровой процесс есть формальное доказательство.

Итак, мы можем уточнить рамочную концепцию алгоритма следующим образом.

Имеется входной алфавит $V = \{a_1, \dots, a_n\}$, и множество входных данных X есть некоторое множество слов в этом алфавите, то есть язык $X \subseteq V^*$, где V^* есть множество всех слов в указанном алфавите. Задан выходной алфавит $W = \{b_1, \dots, b_m\}$, и множество результатов Y есть некоторый язык в выходном алфавите, то есть $Y \subseteq W^*$. Тогда алгоритм A типа (X, Y) , представленный формулой (1), может быть описан в виде

$$A: X \subseteq V^* \xrightarrow{\square} Y \subseteq W^* \quad (2)$$

О таком алгоритме будем говорить, как об алгоритме типа (V, W) , указывая лишь входной и выходной алфавиты. Без существенного ограничения общности можно допустить совпадение входного и выходного алфавитов, а также считать множества входных и выходных слов множествами всех слов в своих алфавитах.

Необходимо заметить, что результат, вообще говоря, определен не для каждого входного слова $x \in X$ (и точка под стрелкой в формулах (1) и (2) есть условное обозначение этого обстоятельства). Если результат работы алгоритма A со словом $x \in X$ определен, то будем говорить, что *алгоритм A применим к слову $x \in X$* и писать $!A(x)$, обозначая сам результат через $A(x)$. Тем самым выходное слово $y = A(x) \in Y$. В противном случае пишем $\neg !A(x)$ и говорим, что *алгоритм A не применим к слову $x \in X$* . Если $(\forall x) !A(x)$, то алгоритм A будем называть **полным алгоритмом типа (V, W)** и писать

$$A: X \subseteq V^* \longrightarrow Y \subseteq W^* \quad (3)$$

(без точки под стрелкой).

Теперь введем понятие вербальной (словарной) функции и вычислимой (в интуитивном смысле слова) вербальной функции.

Вербальная (словарная) функция типа (V, W) есть любое, в общем случае частичное, отображение вида:

$$f: V^* \xrightarrow{\square} W^* \quad (4)$$

(точка под стрелкой указывает на частичность).

Здесь разумно привести некоторые примеры.

Примеры. 1) Функция удвоения: для любого слова $x \in X \subseteq V^*$ значение функции $f(x) = xx = x^2$. Эта функция определена для всех слов во входном алфавите, который совпадает с выходным.

2) Характеристическая функция определенного множества слов в заданном входном алфавите V :

$$f(x) = \begin{cases} 1, & \text{если заданное непустое слово } u \in V^* \text{ входит в } x \\ 0 & \text{иначе} \end{cases}.$$

Здесь мы полагаем, что выходной алфавит $W = \{0,1\}$.

3) В качестве частичной вербальной функции можно рассмотреть следующую. Определим множества $X = Y \subseteq \{0,1\}^*$ как множества двоичных кодов натуральных чисел и положим, что $f(x) = y$ тогда и только тогда, когда $x = y^2$. Очевидно, что функция определена только для полных квадратов.

Для вербальной функции f обозначим $D(f)$ область ее определения.

Следующее понятие является в этом разделе основным.

Вербальная функция f типа (V, W) называется **вычислимой в интуитивном смысле слова**, если существует алгоритм A_f такой, что для любого слова $x \in V^*$ имеет место следующее: $A_f(x)$ тогда и только тогда, когда $x \in D(f)$, и в этом случае $A_f(x) = f(x)$.

То есть указанный алгоритм результативен для всех слов из области определения функции, и только для них, и результат его работы с исходным словом совпадает со значением функции на этом слове. Важно подчеркнуть, что в этом определении не фигурирует какая-либо точная математическая модель алгоритма, и это понятие рассматривается в рамках очерченной выше содержательной концепции. И еще важно обратить внимание на то, что в теории алгоритмов само понятие функции трактуется в ином ключе, чем в классической математике: в теории алгоритмов функция есть не только некоторое соответствие из одного множества в другое, но еще и процесс ее вычисления.

Перейдем теперь к рассмотрению точной модели алгоритма в виде нормального алгорифма Маркова. Заметим сразу, что школе А.А. Маркова принят именно термин «алгорифм», и сочетание «нормальный алгоритм» считается недопустимым.

Нормальный алгорифм Маркова

Прежде всего необходимо заметить, что в нашем курсе (и, соответственно, в этой статье) нормальный алгорифм рассматривается исключительно как точная модель алгоритма, но не как основа дальнейшего построения математики в рамках конструктивного направления [8]. Можно сказать, что мы строим некоторую теоретико-множественную модель нормального алгорифма, определяя это понятие аналогично похожим понятиям в теории формальных языков (например, понятиям грамматики или автомата [9]).

Начнем с некоторых предварительных определений.

Пусть V некоторый алфавит, а x и y слова в этом алфавите.

Говорят, что **слово x входит в слово y** , если существуют такие слова $y_1, y_2 \in V^*$ (возможно пустые), что $y = y_1 x y_2$. Например, слова «ход» и «вход» входят в слово «входит». Соответствующее отношение на множестве слов будем обозначать $\prec$, используя тем самым запись $x \prec y$. Очевидны следующие свойства этого отношения: каждое слово входит само в себя (отношение рефлексивно), пустое слово (обозначаемое далее буквой λ) входит в любое слово, и если $x \prec y$, а $y \prec z$, то $x \prec z$. Тем самым отношение $\prec$ является предпорядком.

Если $x \prec y$, то возникает упорядоченная тройка слов (y_1, x, y_2) , называемая **вхождением слова x в слово y** . В теории нормальных алгорифмов вхождение записывают так: $y_1 * x * y_2$, где «звездочка» является «метасимволом», не принадлежащем алфавиту V . При этом пустое слово часто не пишется вообще, то есть, например,

вхождение слова «вход» в слово «входит» следует записывать так: *вход*ит. Слово y_1 называют **левым**, а слово y_2 **правым крылом** вхождения.

Одно и то же слово может входить в другое слово несколько раз. Например, $x = aba, y = ababab$. Мы имеем тут два вхождения: $*aba*bab$ и $ab*aba*b$. Среди всех вхождений слова x в слово y выделяют первое, или главное. Это то, которое имеет наименьшую длину левого крыла, то есть самое левое. Из показанных выше двух вхождений первое как раз будет главным.

Дадим теперь определение формулы подстановки.

Формула подстановки в алфавите V есть, по определению, упорядоченная пара слов в этом алфавите, записываемая в виде $u \rightarrow v$, где «стрелка» есть опять-таки метасимвол, отделяющий **левую часть формулы** (слово u) от **правой** (слова v). Часто дальше будем говорить просто «формула» вместо «формула подстановки».

Пусть теперь ω есть формула $u \rightarrow v$, а $x \in V^*$. Говорят, что **формула ω применима к слову x** (или является **подходящей для слова x**), если ее левая часть входит в слово: $u \prec x$. Пусть $x_1 * u * x_2$ - первое вхождение u в x . **Результат ωx применения** формулы ω к слову x есть слово $y = \omega x = x_1 v x_2$. Это схематически можно показать так:

$$\begin{array}{ccc} x = x_1 & u & x_2 \\ \downarrow & & \\ y = x_1 & v & x_2 \end{array}$$

Говорят в этом случае, что **слово $y = \omega x$ получено в результате замены первого вхождения левой части формулы ω в слово x на правую**.

Такую операцию над исходным словом называют также **нормальной подстановкой**.

Основное определение: нормальный алгоритм (НА) в алфавите V есть упорядоченная тройка $A = (V, S, P)$, где S - **схема НА**, представляющая собой упорядоченный набор (кортеж, конечная последовательность) формул подстановки в алфавите V , в котором отмечены некоторые формулы, называемые **заключительными** и образующие частичный кортеж (подпоследовательность) P . Заключительная формула в схеме отмечается точкой после стрелки.

Будем далее, если контекст это позволяет, пользоваться аббревиатурой НА, или говорить просто «алгоритм», имея в виду именно нормальный алгоритм Маркова.

Пример схемы:

$$A_0 : \begin{cases} \#a \rightarrow a\# \\ \#b \rightarrow b\# \\ \# \rightarrow \square aba \\ \rightarrow \# \end{cases} \quad (5)$$

Здесь предполагается, что это схема НА в алфавите $V = \{a, b, \#\}$, причем третья сверху формула является заключительной, а левая часть самой нижней формулы есть пустое слово.

Из приведенного примера можно понять синтаксис записи схем нормальных алгоритмов: формулы схемы пишутся в столбик и упорядочиваются сверху вниз, схема окаймляется левой фигурной скобкой и предваряется обозначением НА с двоеточием.

В общем случае схема НА выглядит так:

$$A: \begin{cases} u_1 \rightarrow [\Box]v_1 \\ u_2 \rightarrow [\Box]v_1 \\ \vdots \\ u_n \rightarrow [\Box]v_n \end{cases}$$

Квадратные скобки означают необязательность вхождения точки.

Так строится «статическая» часть определения нормального алгорифма. Теперь нужно определить «динамику», то есть точно определить процесс преобразования слов, задаваемый нормальным алгорифмом. Есть смысл сначала дать содержательное описание этого процесса, называемого *процессом работы НА с исходным словом*.

Пусть задан НА и входное слово (в каком-то алфавите).

Схема НА просматривается сверху вниз в поисках первой применимой ко входному слову формулы. Если таковой не оказывается, то процесс заканчивается и само входное слово считается его результатом (то есть НА вычисляет тогда тождественную функцию). Иначе, найденная формула применяется, как описано выше (операция нормальной подстановки). Если эта формула оказалась простой (не заключительной), то с полученным в результате ее применения словом поступают так же, как с исходным. Если же она оказалась заключительной, то после ее применения процесс заканчивается и слово, полученное в результате ее применения, считается результатом. Таким образом, процесс продолжается до тех пор, пока не произойдет одно из двух событий: 1) *естественный обрыв*, когда в схеме нет ни одной применимой к текущему слову формулы; в этом случае говорят, что слово не поддается данному НА (или его схеме), и оно считается результатом; 2) к текущему слову была применена заключительная формула, и тогда результатом считается слово, полученное применением этой формулы к текущему. Может оказаться, что ни одним из шагов не выполняется условие останова (прекращения процесса). В этом случае результат не определен, и говорят, что данный НА не применим ко входному слову.

Чтобы дать точное (формальное) определение процесса работы, нужно ввести некоторые обозначения и определения.

1) Будем писать $A: x \mapsto y$ и говорить, что **НА А непосредственно просто переводит слово x в слово y** , если $y = \omega x$, где ω - первая (самая верхняя) формула в схеме НА, оказавшаяся простой.

2) Будем писать $A: x \mapsto \Box y$ и говорить, что **НА А непосредственно заключительно переводит слово x в слово y** , если $y = \omega x$, где ω - первая (самая верхняя) формула в схеме НА, оказавшаяся заключительной.

3) Будем писать $A: x \Rightarrow y$ и говорить, что **НА А просто переводит слово x в слово y** (без слова «непосредственно!»), если существует последовательность слов $x = x_0, x_1, \dots, x_n = y, n > 0$, в которой для каждого $i = \overline{0, n-1}$ выполняется $A: x_i \mapsto x_{i+1}$.

4) Будем писать $A: x \Rightarrow \Box y$ и говорить, что **НА А заключительно переводит слово x в слово y** , если $A: x \mapsto \Box y$ или существует такое слово z , что $A: x \Rightarrow z$ и $A: z \mapsto \Box y$.

Можно сказать, что в пп. (1) и (2) дано формальное определение *шага процесса работы НА со словом*, а в пп. (3) и (4) определяется последовательность таких шагов (в том числе и один шаг).

5) Будем писать $\Box A(x)$, когда слово x **не поддается** НА А.

Теперь мы можем дать строгое определение процесса работы НА со словом.

Процесс работы НА А со словом x есть конечная или бесконечная последовательность слов $x = x_0, x_1, \dots, x_n, \dots$, где $n \geq 0$, причем для каждого $i \geq 0$ $A: x_i \mapsto x_{i+1}$ или

$A: x_i \mapsto \square x_{i+1}$, если слово x_{i+1} определено в последовательности. Это слово, как и все последующие, считается не определенным, если $A: x_{i-1} \mapsto \square x_i$, то есть последнее слово получено применением заключительной формулы, или $\square A(x_i)$, то есть последнее слово не поддается НА и имеет место естественный обрыв процесса работы.

Если процесс работы НА A со словом x конечен, то его последнее слово обозначается $A(x)$ и называется **результатом работы НА A со словом x** . В этом случае говорят, что **НА A применим к слову x** и пишут $!A(x)$. В противном случае говорят, что **НА A не применим к слову x** и пишут $\neg !A(x)$.

Чтобы завершить описание основной модели алгоритма (НА), необходимо определить понятие вербальной функции, вычислимой по Маркову и сформулировать основную гипотезу, утверждающую универсальность предложенной модели.

Вербальная функция f типа (V, V) называется **вычислимой по Маркову**, если может быть построен НА A_f такой, что для любого слова $x \in V^*$ имеет место следующее: $!A_f(x)$ тогда и только тогда, когда $x \in D(f)$, и в этом случае $A_f(x) = f(x)$.

Необходимо обратить внимание, что в этом определении вместо слова «существует» (в определении интуитивной вычислимости) стоит «можно построить». Это очень важно: чтобы доказать вычислимость по Маркову, нужно предъявить схему соответствующего НА и доказать, что этот НА вычисляет заданную функцию, то есть доказательство должно быть конструктивным. Конечно, в любом случае, при разработке алгоритма необходимо его точное описание и доказательство корректности, но всё равно тогда возникает некое уточнение интуитивного понимания алгоритма, например, в виде псевдокода или даже программы на языке высокого уровня. То есть всякий раз, когда от интуитивной концепции вычислимости мы переходим к той или иной точной модели, доказательство вычислимости функции обязано быть конструктивным. Это, повторим, важнейший аспект в методике преподавания теории алгоритмов студентам-программистам.

Основная гипотеза теории алгоритмов в версии теории нормальных алгорифмов Маркова называется **принципом нормализации** и состоит в следующем:

Всякая вербальная функция, вычисляемая в интуитивном смысле слова, вычислима по Маркову.

Подчеркнем, что это именно гипотеза, а не теорема, так как в формулировке фигурирует не определенное математически точно понятие функции, вычислимой в интуитивном смысле слова.

На этом описание основной модели закончено.

Рассмотрим некоторые примеры нормальных алгорифмов, важные для дальнейшего изложения.

Примеры

1) **Тождественный НА Id** , задаваемый схемой

$\{\rightarrow\}$.

любое слово в заданном алфавите V перерабатывает в это же слово за один шаг, то есть

$$(\forall x \in V^*)(Id : x \mapsto \square x).$$

2) **Нигде не применимый НА $Null$** , задаваемый схемой

$\{\rightarrow\}$

Очевидно, что для любого входного слова процесс работы такого НА с ним будет бесконечным, то есть $(\forall x \in V^*) \neg !Null(x)$

3) **НА левого присоединения фиксированного слова.**

НА LC , заданный схемой

$$\{\rightarrow \cdot u,$$

где u - фиксированное слово в алфавите V , вычисляет функцию левого присоединения к любому слову в алфавите V данного слова u , то есть

$$(\forall x \in V^*)(LC: x \mapsto \sqcup ux)$$

4) НА правого присоединения фиксированного слова.

НА Rc , заданный схемой

$$\begin{cases} \# \eta \rightarrow \eta \#, \eta \in V \\ \# \rightarrow \cdot u \\ \rightarrow \# \end{cases} \quad (6)$$

в алфавите $V_1 = V \cup \{\#\}$, где $\# \notin V$, вычисляет функцию правого присоединения данного фиксированного слова u в алфавите V к любому слову в алфавите V , т.е.

$$(\forall x \in V^*)(Rc: x \Rightarrow \sqcup xu)$$

Действительно, к произвольному слову $x \in V^*$ применима только самая нижняя формула в схеме алгоритма Rc . После ее применения оказывается применимой одна из формул верхней строки, если слово x не пусто. Эти формулы применяются до тех пор, пока «решетка» ($\#$) не окажется последней буквой очередного слова, выводимого из исходного слова x . Тогда она посредством применения средней формулы схемы заключительно заменяется на слово u .

Заметим, что в этой схеме фигурирует параметр η , пробегающий алфавит V , и, тем самым, верхняя строка схемы представляет конечное множество формул, число которых равно числу букв алфавита V . В схему НА можно вводить параметры, но необходимо, чтобы он пробежал некоторое конечное множество. Схема НА Rc , как нетрудно видеть, обобщает схему (5).

5) НА удвоения

Рассмотрим НА $Double$, задаваемый схемой в алфавите $V_1 = V \cup \{\alpha, \beta\}$, причем $\alpha, \beta \notin V$:

$$\begin{cases} \alpha \xi \rightarrow \xi \beta \xi \alpha, \xi \in V \\ \beta \xi \eta \rightarrow \eta \beta \xi, \eta, \xi \in V \\ \beta \rightarrow \\ \alpha \rightarrow \cdot \\ \rightarrow \alpha \end{cases} \quad (7)$$

Можно показать, что $(\forall x \in V^*)(Double: x \Rightarrow \sqcup xx = x^2)$, т.е. этот алгоритм удваивает любое входное слово в алфавите V .

Мы в рамках этой работы, ради экономии объема, не анализируем подробно работу приведенных в примерах нормальных алгоритмов. Подробности такого рода будут рассмотрены в отдельной публикации, специально посвященной проблемам техники разработки схем нормальных алгоритмов.

Можно заметить также, что если вместо формулы $\beta \rightarrow$ поставить формулу $\beta \xi \alpha \rightarrow \xi \alpha$, то копия будет инвертирована, т.е. получится слово xx^R .

Если схему (7) модифицировать следующим образом

$$Double^{\#} : \begin{cases} \alpha\xi \rightarrow \xi\beta\xi\alpha \\ \beta\xi\eta \rightarrow \eta\beta\xi \\ \alpha \rightarrow \$ \\ \beta\xi\# \rightarrow \#\xi \\ \# \rightarrow \cdot\# \\ \rightarrow \alpha \end{cases} \quad (8)$$

$$(\xi, \eta \in V; \alpha, \beta, \# \notin V)$$

то получим алгоритм удвоения через разделитель, то есть $Double^{\#}(x) = x\#x$.

Эквивалентность нормальных алгоритмов. Теорема о переводе

НА A и B типа (V, V) над алфавитом V называют **вполне эквивалентными относительно алфавита V** , если для всякого слова $x \in V^*$ имеет место $!A(x) \Leftrightarrow !B(x)$ и если выполняется $!A(x)$, то $A(x) = B(x)$.

В теории нормальных алгоритмов используется выражение $A(x) \sqsubseteq B(x)$, называемое **условным равенством**, смысл которого состоит в том, что левая часть его определена тогда и только тогда, когда определена правая, и в этом случае они обозначают одно и то же слово. Это не что иное, как записанное для слова x требование полной эквивалентности НА A и B , т.е. НА A и B вполне эквивалентны относительно алфавита V тогда и только тогда, когда $(\forall x \in V^*)(A(x) \sqsubseteq B(x))$.

Рассмотрим теперь некоторые преобразования схем НА, которые дают НА, вполне эквивалентный исходному относительно заданного алфавита.

1) Замыкание нормального алгоритма

Так называется НА, схема которого получается из схемы исходного НА путем добавления после всех формул заключительной формулы $\rightarrow \square$. Замыкание НА A обозначается $A^{\square}$. Легко показать, что эти НА вполне эквивалентны относительно алфавита, в котором задан исходный НА.

Переход к замыканию позволяет впредь при обсуждении применимости НА к слову не рассматривать ситуацию естественного обрыва, то есть считать, что применимость НА к слову означает, что на последнем шаге процесса работы применяется заключительная формула.

2) Естественное и формальное распространение

Для НА A в алфавите V можно той же самой схемой задать НА A' в более широком алфавите (т.е. подавать на его вход слова в некотором расширении алфавита V). Тогда очевидно, что для любого слова x в алфавите V имеет место $A(x) \sqsubseteq A'(x)$. Таким образом, НА A и A' вполне эквивалентны относительно алфавита V . НА A' называют при этом **естественным распространением** исходного на более широкий алфавит.

Формальное распространение A^f НА A на более широкий алфавит $V' \supset V$ определяется тем, что к схеме A , в начале ее («сверху»), приписываются формулы $\eta \rightarrow \eta$ для всех букв η из $V \setminus V$. Тогда, как нетрудно видеть, $(\forall x \in V^*)(A^f(x) \sqsubseteq A(x))$, но, в отличие от естественного распространения, имеет место $(\forall y \in V \setminus V)(\neg !A^f(y))$, то есть НА A^f не применим ни к одному из слов, не являющимся словом в алфавите V .

Следующее, более сложное, преобразование НА связано с так называемой теоремой о переводе, которая ограничивает число дополнительных букв при построении НА, вычисляющего некоторую функцию типа (V, V) для некоторого произвольно за-

данного алфавита V (например, в рассмотренных выше НА удвоения и инвертирования потребовались две дополнительные буквы). Оказывается, что именно двумя буквами можно ограничить расширение алфавита в любом случае. При этом доказывалось также, что существует такая вербальная функция указанного типа, которую невозможно вычислить посредством НА, не расширяя алфавит.

Чтобы сформулировать соответствующий результат, введем понятие перевода в двухбуквенный алфавит.

Пусть $V = \{a_1, \dots, a_n\}$ - исходный алфавит, а алфавит $V_\alpha = \{\alpha, \beta\}$ состоит из двух букв. Для простоты будем считать, что эти два алфавита не пересекаются (хотя это требование не является обязательным).

Тогда назовем переводом буквы $a_k \in V$ в алфавит V_α слово $[a_k] = \alpha\beta^k\alpha$. Перевод непустого слова $x = x(1)x(2)\dots x(m) \in V^*$ есть, по определению, соединение переводов его букв:

$$[x] = [x(1)][x(2)]\dots[x(m)].$$

Перевод пустого слова естественно определить как пустое слово. Например, если $V = \{a, b, c\}$, и $x = abca$, то $[x] = \alpha\beta\alpha\alpha\beta\beta\alpha\alpha\beta\beta\alpha\alpha\beta\alpha$. Здесь предполагается, что буквы исходного алфавита нумеруются, как 1-я, 2-я и 3-я.

Часто в качестве двухбуквенного алфавита, в который делается перевод, берется алфавит $V_0 = \{0, 1\}$, в котором «ноль» и «единица» фигурируют только как буквы, но не как числа. Понятие натурального числа появится позже.

Сформулируем теперь без доказательства теорему, которая в данном контексте называется теоремой о переводе, хотя она на самом деле является одним из следствий более общего утверждения, принадлежащего А.А. Маркову и названной им теоремой о переводе [7].

Теорема 1. Каков бы ни был нормальный алгоритм A типа (V, V) над алфавитом V , может быть построен нормальный алгоритм B в алфавите $V \cup V_\alpha$, вполне эквивалентный A относительно алфавита V , то есть для любого слова $x \in V^*$ имеет место условное равенство $A(x) \sqsubseteq B(x)$.

Идея доказательства состоит в том, что все буквы из исходного расширения алфавита V заменяются их переводами в алфавит V_α , соответственно преобразуется схема исходного НА и все слова его процесса работы с исходным словом, после чего доказывалась полная эквивалентность полученного НА и исходного.

Теоремы сочетания

Алгоритм (в интуитивном смысле) может быть «запрограммирован» не только в виде нормального алгоритма - выписыванием его схемы, но через определенные правила комбинации (сочетания) уже построенных нормальных алгоритмов.

Например, следующее предписание определяет точный алгоритм переработки слов в заданном алфавите:

- 1) применить к исходному слову x нормальный алгоритм A ;
- 2) если слово $A(x)$ определено, то к нему применить нормальный алгоритм B ;
- 3) результатом считать слово $B(A(x))$, если оно определено.

Это предписание не является априори нормальным алгоритмом, поскольку оно не задано единой схемой такого алгоритма, а оперирует нормальными алгоритмами, словно некоторыми «блоками». Оно определяет один из способов сочетания нормальных алгоритмов, называемый *композицией* (иногда *последовательной композицией*).

Оказывается, что композицию и другие сочетания нормальных алгоритмов, которые будут рассмотрены ниже, можно «запрограммировать» в виде схем нормальных

алгоритмов. Теоремы о возможности построения таких схем называются **теоремами сочетания**.

Мы рассмотрим в этом разделе только два способа сочетания из четырех: композицию и разветвление, так как они необходимы нам для доказательства основной теоремы. Объединение (независимое, «параллельное», применение нормальных алгоритмов ко входному слову) и повторение («моделирующее» обычный программный цикл) обсудим в отдельных публикациях. Теоремы сочетания здесь формулируются без доказательства. Для простоты считаем, что все сочетаемые нормальные алгоритмы заданы в одном и том же алфавите V .

Сформулируем теорему композиции.

Теорема 2 (о композиции нормальных алгоритмов). Каковы бы ни были нормальные алгоритмы A и B в алфавите V , может быть построен такой нормальный алгоритм C над алфавитом V , что для любого слова $X \in V^*$ имеет место условное равенство $C(x) \sqsubseteq B(A(x))$.

Нам будет удобно дальше обозначать НА C как $B \circ A$, то есть $B \circ A(x) \sqsubseteq B(A(x))$.

Способ сочетания нормальных алгоритмов, называемый **разветвлением**, задается так:

- 1) к исходному слову x применить алгоритм C ;
- 2) если слово $C(x)$ существует и равно пустому слову, то к слову x применить алгоритм A и слово $A(x)$ считать общим результатом;
- 3) если слово $C(x)$ существует и не равно пустому слову, то к слову x применить алгоритм B и считать слово $B(x)$ общим результатом;
- 4) если слово $C(x)$ не определено, то не определен и общий результат.

Теорема о «программировании» разветвления нормальных алгоритмов в виде схемы некоторого нормального алгоритма называется теоремой о разветвлении и формулируется следующим образом.

Теорема 3 (о разветвлении нормальных алгоритмов). Каковы бы ни были нормальные алгоритмы A, B и C в алфавите V , может быть построен алгоритм D над алфавитом V такой, что для любого слова $x \in V^*$ выполняется следующее: 1) если $\neg !C(x)$ то $\neg !D(x)$; 2) если $!C(x)$, то $D(x) = A(x)$ при $C(x) = \lambda$ и $D(x) = B(x)$ при $C(x) \neq \lambda$.

Нормальный алгоритм D , который может быть построен согласно теореме о разветвлении, будем обозначать $_C(A \vee B)$ и называть C -**разветвлением алгоритмов** A и B . Коротко этот алгоритм можно задать такой записью:

$$_C(A \vee B)(x) \sqsubseteq \begin{cases} A(x), \text{ при } C(x) = \lambda \\ B(x), \text{ при } C(x) \neq \lambda \end{cases}.$$

В программистских терминах C -разветвление можно описать так:

if $C(x) = \lambda$ **then** $y := A(x)$ **else** $y := B(x)$.

Сформулированные здесь теоремы о композиции и разветвлении будут использованы при доказательстве основного результата о неразрешимости проблемы применимости для нормальных алгоритмов.

Универсальный нормальный алгоритм

Приступая к объяснению конструкции универсального НА, необходимо заметить следующее.

Очевидно, что наши действия по применению НА, заданного своей схемой, являются шагами некоторого алгоритма (в интуитивном смысле слова). Следовательно,

согласно принципу нормализации, их можно задать некоторым нормальным алгоритмом. Он должен как-то воспринимать на входе любой НА и работать за него. Но для того, чтобы одни алгоритмы использовать в качестве исходных данных для других, необходимо алгоритм как данное, *превратить в конструктивный объект*, то есть представить в виде слова в некотором алфавите. Существует два основных способа такой *конструктивизации* нормального алгоритма: посредством *изображения* и посредством *записи*.

Пусть НА A является алгоритмом в алфавите V . *Изображение НА A* есть слово в алфавите $V \cup \{\alpha, \beta, \gamma\}$, где $\alpha, \beta, \gamma \notin V$ которое строится по схеме A следующим образом: формулы выписываются в порядке следования их в схеме, причем буква γ служит разделителем между формулами, каждая стрелка заменяется буквой α , а каждая точка - буквой β .

Например, алгоритм Lc левого присоединения слова $u \in V^*$, заданный схемой $\{\rightarrow \cdot u,$

будет иметь изображение: $\alpha\beta u$.

Алгоритм Rc правого присоединения непустого слова $u \in V^*$, заданный схемой в алфавите $V \cup \{\#\}$, где $\# \notin V$, имеющей вид (в сокращенной записи)

$$\left\{ \begin{array}{l} \rightarrow \# \\ \# \eta \rightarrow \eta \#, \eta \in V \\ \# \rightarrow \cdot u \end{array} \right.$$

будет иметь такое изображение: $\alpha\#\gamma\#a_1\alpha a_1\#\gamma\ldots\gamma\#a_n\alpha a_n\#\gamma\#\alpha\beta u$, где предполагается, что $V = \{a_1, \dots, a_n\}$.

В общем случае алгоритм, задаваемый схемой

$$\left\{ \begin{array}{l} u_1 \rightarrow [\square]v_1 \\ \dots \\ u_n \rightarrow [\square]v_n \end{array} \right.$$

в алфавите V , будет иметь изображение в виде слова $u_1\alpha[\beta]v_1\gamma\ldots\gamma u_n\alpha[\beta]v_n$.

Вместо изображения нормального алгоритма часто используют *запись нормального алгоритма* - слово, определяемое как перевод изображения в алфавит $V_0 = \{0, 1\}$. При построении записи НА в алфавите $V = \{a_1, \dots, a_n\}$ принято считать, что буквы α, β, γ , фигурирующие в изображении, получают номера $n+1, n+2, n+3$ соответственно.

Так, запись алгоритма левого присоединения слова $abca$ в алфавите $V = \{a, b, c\}$ будет иметь вид 011110111111001001100111010.

$\alpha \quad \beta \quad a \quad b \quad c \quad a$

Заметим, что в некоторых источниках буква γ в изображении НА пишется и в конце, то есть пишется сразу справа от изображения каждой формулы подстановки.

Запись нормального алгоритма A будем обозначать $\langle A \rangle$.

Нетрудно понять, что как по изображению, так и по записи схема нормального алгоритма восстанавливается однозначно. Использование записи удобно в теоретических рассуждениях, так как позволяет считать все нормальные алгоритмы представленными словами в одном и том же двухбуквенном алфавите (своего рода «двоичным кодом»).

Теорема 4 (теорема об универсальном нормальном алгоритме). Для любого алфавита V и любого НА A в этом алфавите может быть построен НА U над алфавитом $V \cup V_0 \cup \{\#\}$ такой, что для любого слова $x \in V^*$ имеет место условное равенство $U(\langle A \rangle \# x) \sqsubseteq A(x)$.

Алгоритм U , построенный согласно данной теореме, называют **универсальным нормальным алгоритмом**.

Существует вариант формулировки теоремы об универсальном алгоритме, в которой вместо записи фигурирует изображение «объектного» алгоритма A .

Доказательство теоремы 4 чрезвычайно сложно и не может быть здесь рассмотрено. В курсе теории алгоритмов для студентов-программистов эта теорема также только формулируется.

Содержательный смысл теоремы состоит в том, что универсальный алгоритм получает на вход пару «объектный алгоритм - слово», декодирует запись объектного алгоритма, восстанавливая тем самым его схему, и применяет эту схему согласно известным правилам ко входному слову. Тем самым заданный в определении нормального алгоритма *алгоритм* (определенный вне использования нормальных алгоритмов) применения схемы, согласно теореме об универсальном алгоритме сам может быть запрограммирован как *нормальный алгоритм*.

С программистской же точки зрения универсальный алгоритм есть не что иное как *универсальный интерпретатор* в классе программ, записанных в виде схем нормальных алгоритмов. Эту аналогию как раз важно провести в аудитории студентов-программистов. А именно, если пара (объектная программа P , данные для нее d), то системная программа-интерпретатор Int , «декодируя» определенным образом объектную программу, сразу применяет ее к исходным данным. Это можно записать в виде условного равенства:

$$Int(P, d) \sqsubseteq P(d)$$

То есть интерпретатор выполняет функцию универсального алгоритма в некотором классе объектных программ.

Разрешимые и перечислимые множества (языки)

Язык (множество слов) $L \subseteq V^*$ называют **алгоритмически разрешимым**, если может быть построен нормальный алгоритм A_L над алфавитом V такой, что для каждого $x \in V^*$ имеет место $\neg A_L(x)$ и $A_L(x) = \lambda \Leftrightarrow x \in L$.

Алгоритм A_L , фигурирующий в вышеприведенном определении, называют **разрешающим алгоритмом** для языка L .

Легко показать, что, например, множество слов, содержащих хотя бы одно вхождение заданного непустого слова, будет алгоритмически разрешимо.

От ситуации возможности построения разрешающего НА для языка следует отличать ситуацию **полуразрешающего** НА. Так называется НА $\tilde{A}_L$, применимый к слову $x \in V^*$ тогда и только тогда, когда $x \in L$. Принципиальное отличие от разрешающего НА состоит в том, что разрешающий всегда заканчивает работу за конечное число шагов, и мы рано или поздно получим ответ на вопрос о принадлежности слова языку. Что же касается полуразрешающего НА, то неизвестно, остановится он когда-нибудь и, следовательно, будет ли вообще дан ответ на вопрос о принадлежности.

Интуитивно очевидно, что из невозможности построения полуразрешающего НА следует невозможность построения и разрешающего. Докажем это строго.

Теорема 5. Если для языка $L \subseteq V^*$ невозможен полуразрешающий НА, то невозможен и разрешающий.

Доказательство. Предположим противное, то есть предположим, что для языка $L \subseteq V^*$ не может быть построен полуразрешающий НА, но возможно построение разрешающего. Пусть A_L - такой НА. Тогда по теореме разветвления (теорема 3), построим НА

$$B_L =_{A_L} (A_L \vee \text{Null}) .$$

Ясно, что для любого слова $x \in V^*$ имеет место $!B_L(x) \Leftrightarrow x \in L$, и НА B_L является полуразрешающим вопреки предположению.

Образно говоря, разрешающий алгоритм можно «испортить», сделав его полуразрешающим.

Определим теперь понятие перечислимого множества (языка). Здесь нам понадобится понятие **конструктивного натурального числа** (КНЧ). По определению, это любое слово в алфавите V_0 вида $011\dots 1 = 01^n, n \geq 0$.

Язык (множество слов) $L \subseteq V^*$ называют **алгоритмически перечислимым**, если существует нормальный алгоритм N_L над алфавитом $V \cup V_0$ такой, что для каждого натурального числа n имеет место $!N_L(n)$, $N_L(n) \in L$ и для каждого $x \in L$ осуществимо такое натуральное число n , что $N_L(n) = x$.

В дальнейшем будем говорить просто о разрешимых и перечислимых языках.

Понять смысл определения перечислимого языка будет проще, если вспомнить классическое теоретико-множественное понятие нумерации множества.

Именно, под **нумерацией множества** M понимается любое сюръективное отображение $\nu: \mathbb{N} \rightarrow M$ множества натуральных чисел на множество M . Для простоты будем рассматривать далее только биективные (взаимно однозначные) нумерации. Можно показать, что это не ограничивает существенно общности. Множество M в таком случае есть не что иное, как счетное множество.

Алгоритм N_L называют при этом **алгоритмом, перечисляющим множество** L (или просто **перечисляющим алгоритмом**, если множество подразумевается). Говорят также, что **алгоритм** N_L **перечисляет множество** L . В свете теоретико-множественного понятия нумерации, перечислимое множество есть теоретико-алгоритмический (конструктивный) аналог понятия счетного множества, перечисляющий НА есть нумерация, заданная в виде нормального алгоритма, а осуществимость КНЧ n в конце определения есть возможность задать в виде НА отображение, обратное нумерации (если последняя биективна).

Пример. Запишем НА, который перечисляет множество конструктивных целых чисел (КЦЧ). КЦЧ - это слово вида $[-]n = [-]011\dots 1$. Напишем сначала формулу обычной нумерации $\nu: \mathbb{N} \rightarrow \mathbb{N}$ множества целых чисел:

$$\nu(n) = \begin{cases} -\frac{n}{2}, & \text{если } n \text{ четное} \\ \frac{n+1}{2}, & \text{если } n \text{ нечетное} \end{cases}$$

Соответствующий нормальный алгоритм строим по теореме разветвления:

$$N =_C (A \vee B),$$

где $C: \begin{cases} 11 \rightarrow \\ 0 \rightarrow \square \end{cases}$ - НА проверки четности,

$$\begin{aligned}
 A: & \begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \square \\ 01 \rightarrow -0\alpha \\ 0 \rightarrow \square 0 \end{cases} \quad - \text{НА деления пополам с переходом к противоположному числу } (\alpha \notin V_0), \\
 B: & \begin{cases} \alpha 11 \rightarrow 1\alpha \\ \alpha \rightarrow \square \\ 01 \rightarrow 0\alpha 1 \end{cases} \quad - \text{НА прибавления единицы и последующего деления пополам.}
 \end{aligned}$$

Может быть доказана следующая теорема:

Теорема 6. Всякое алгоритмически разрешимое множество алгоритмически перечислимо.

Заметим, что обратное неверно, что будет видно из дальнейших рассмотрений.

Легко понять также, что дополнение разрешимого языка также разрешимо.

В контексте проблемы применимости особенно важна характеристика перечислимых языков через области применимости нормальных алгоритмов.

Пусть дан НА A над алфавитом V . **Областью применимости** НА A называется множество $M_A^V = \{x : !A(x), x \in V^*\}$.

Теорема 7. Язык $L \subseteq V^*$ перечислим тогда и только тогда, когда он является областью применимости относительно алфавита V некоторого нормального алгоритма.

Доказательство теорем 6 и 7 мы не приводим (см., например, [8]).

Проблема применимости как неразрешимая массовая проблема

Проблема применимости для нормальных алгоритмов формулируется как частная и как общая.

Начнем с частной проблемы.

Дан нормальный алгоритм A в алфавите V . Можно ли построить НА B над алфавитом V так, чтобы он был применим к любому слову $x \in V^*$, причем $B(x) = \lambda$ тогда и только тогда, когда $\neg !A(x)$.

Другими словами, НА B должен быть разрешающим для дополнения области применимости НА A относительно алфавита V .

Общая проблема применимости формулируется следующим образом: можно ли построить нормальный алгоритм B над алфавитом $V \cup \{0,1\}$ такой, что для любых нормального алгоритма A в алфавите V и слова $x \in V^*$ выполнялось $!B(\langle A \rangle \# x)$ и $!B(\langle A \rangle \# x) = \lambda \Leftrightarrow \neg !A(x)$, где буква $\#$ не принадлежит алфавиту $V \cup \{0,1\}$.

Можно сказать, что если в постановке частной проблемы применимости ставится вопрос о «тесте» на применимость заданного НА, то в постановке общей проблемы речь идет уже о возможности построения универсального «теста» на применимость в некотором классе нормальных алгоритмов. Ниже будет доказана основная теорема, в доказательстве которой предлагается способ построения НА с неразрешимой частной проблемой применимости. Так что ответ на оба поставленных выше вопроса оказывается отрицательным.

Для этого рассмотрим сначала другую проблему, которая называется **проблемой самоприменимости** и формулируется следующим образом: можно ли построить такой алгоритм B над алфавитом V_0 , что для любого алгоритма A в алфавите $V_1 = V \cup V_0 \cup \{\alpha, \beta\}$, где $\alpha, \beta \notin V \cup V_0$ имеет место $!B(\langle A \rangle)$ и $B(\langle A \rangle) = \lambda \Leftrightarrow \neg !A(\langle A \rangle)$. За-

метим, что согласно теореме о переводе в алфавите V_1 может быть задан любой алгоритм типа $(V \cup V_0, V \cup V_0)$. Следует заметить также, что при обсуждении проблемы самоприменимости мы предполагаем, что все алгоритмы, заданные в каком-то заранее заданном алфавите V заменяются их естественными распространениями на алфавит $V \cup V_0$, чтобы можно было корректно их применять к собственной записи.

Таким образом, алгоритм B должен аннулировать (перерабатывать в пустое слово) записи тех и только тех алгоритмов в алфавите V_1 , которые *не применимы к собственной записи*. Такие алгоритмы называют **несамоприменимыми**. Другими словами, НА B должен быть разрешающим для языка записей несоприменимых нормальных алгоритмов.

Простыми примерами самоприменимого и несоприменимого НА могут служить НА R_c правого присоединения фиксированного слова в заданном алфавите V , естественно распространенный на алфавит $V' \cup V_0$, где $V' = V \cup \{\#\}$, $\# \notin V$ и НА, служащий формальным распространением R_c на алфавит $V' \cup V_0$ соответственно.

Лемма 1. Невозможен нормальный алгоритм B в алфавите $V \cup V_0$, применимый к записи произвольного нормального алгоритма A в этом алфавите тогда и только тогда, когда алгоритм A несоприменим.

Доказательство. Пусть такой алгоритм B построен. Так как и он является алгоритмом в алфавите $V \cup V_0$, то может быть поставлен вопрос о его самоприменимости. Если $!B(\langle B \rangle)$, то поскольку B должен быть применим только к записям несоприменимых алгоритмов, имеет место $\neg !B(\langle B \rangle)$. Но тогда, так как B несоприменимый алгоритм, то должно быть $!B(\langle B \rangle)$. Полученное противоречие доказывает, что алгоритм B не может быть построен.

Таким образом, если «тест» и «тестируемый» на самоприменимость алгоритм заданы в одном и том же алфавите, то проблема самоприменимости неразрешима, так как нельзя построить для языка записей несоприменимых алгоритмов даже полуразрешающий НА (см. теорему 5).

Можно подумать, что, позволив «тесту» (НА B) работать в каком угодно расширении алфавита V_0 , проблему можно решить. Но это не так.

Теорема 8. Невозможен алгоритм над алфавитом V_0 , применимый к записям тех и только тех алгоритмов в алфавите V_1 , которые несоприменимы.

Доказательство. Пусть построен алгоритм B над алфавитом V_0 так, что для всякого алгоритма A в алфавите V_1 имеет место $!B(\langle A \rangle) \Leftrightarrow \neg !A(\langle A \rangle)$. По теореме о переводе тогда может быть построен алгоритм B_1 в алфавите $V_0 \cup \{\alpha, \beta\}$ так, что для любого слова $x \in V_0^*$ имеет место условное равенство $B_1(x) \sqcap B(x)$.

Пусть теперь алгоритм B_2 определен как естественное распространение алгоритма B_1 на алфавит V_1 . Тогда оказывается, что этот алгоритм в алфавите V_1 таков, что для любого алгоритма A в алфавите V_1 имеет место $!B_2(\langle A \rangle) \Leftrightarrow \neg !A(\langle A \rangle)$, что невозможно в силу доказанной леммы.

Суть изложенного доказательства состоит в том, что алгоритм-«тест» и алгоритм, «тестируемый» на самоприменимость, приведены к одному и тому же алфавиту, и мы попадаем в условия леммы 1.

Таким образом, проблема самоприменимости для нормальных алгоритмов, заданных в произвольном наперед заданном алфавите неразрешима.

Здесь методически целесообразно разъяснить учащимся, что алгоритмическая неразрешимость какой-либо проблемы не означает, что на вопрос, который эта проблема ставит, не может быть дан ответ ни в каком конкретном случае. Вполне можно привести пример конкретного нормального алгорифма, для которого может быть доказана его самоприменимость (или несамоприменимость). Но доказательство не есть алгоритм. *Неразрешимость проблемы состоит именно в невозможности построить алгоритм, который для любого ее частного случая дает ответ на вопрос, сформулированный в проблеме.* Именно так следует понимать массовость любой алгоритмически неразрешимой проблемы.

Опираясь на факт неразрешимости проблемы самоприменимости, можно доказать возможность построения нормального алгорифма с неразрешимой частной проблемой применимости, откуда будет следовать, что и общая проблема подавно неразрешима.

Теорема 9. Можно построить нормальный алгорифм A в алфавите $V_2 = \{0, |, \alpha, \beta\}$ так, что невозможен нормальный алгорифм B над алфавитом V_0 , удовлетворяющий условию $B(x) \Leftrightarrow \neg A(x)$ для любого слова $x \in V_0^*$.

Доказательство. Пусть $\# \notin V_2$. По теореме об универсальном алгорифме построим алгорифм U над алфавитом V_2 так, что для любого алгорифма D в алфавите V_2 и любого слова $y \in V_2^*$ имеет место $U(\langle D \rangle \# y) \sqcap D(y)$.

Теперь построим алгорифм U_1 над алфавитом V_2 так, что для любого слова $y \in V_2^*$ $U_1(y) \sqsubseteq U(y \# y)$. (Это можно сделать в виде композиции алгорифма, перерабатывающего всякое слово $y \in V_2^*$ в слово $y \# y$ и алгорифма U , то есть $U_1 = U \circ Double^\#$, см. схему(8)). Следующее место в доказательстве является наиболее тонким.

Мы имеем НА U_1 над алфавитом V_2 , но последний есть расширение алфавита V_0 и, следовательно, по теореме о переводе, может быть заменен вполне эквивалентным ему относительно алфавита V_0 алгорифмом U_2 в двухбуквенном расширении алфавита V_0 , то есть в алфавите V_2 , и для любого слова $x \in V_0^*$ будем иметь: $U_1(x) \Leftrightarrow U_2(x)$. Утверждается, что это и есть искомый и указанный в условии теоремы алгорифм A с неразрешимой частной проблемой применимости, то есть $A = U_2$.

Действительно, пусть построен алгорифм B над алфавитом V_0 так, что $(\forall x \in V_0^*)(B(x) \Leftrightarrow \neg U_2(x))$. Тогда для любого алгорифма D в алфавите V_2 будем иметь

$$B(\langle D \rangle) \Leftrightarrow \neg U_2(\langle D \rangle) \Leftrightarrow \neg U_1(\langle D \rangle) \Leftrightarrow \neg U(\langle D \rangle \# \langle D \rangle) \Leftrightarrow \neg D(\langle D \rangle),$$

То есть НА B оказывается полуразрешающим для проблемы самоприменимости для нормальных алгорифмов в алфавите V_2 , что невозможно в силу теоремы 8.

Итак, проблема применимости для нормальных алгорифмов является неразрешимой.

Теорема 10. Существует перечислимое множество, не являющееся разрешимым.

Эта теорема является следствием доказанной неразрешимости проблемы применимости. Действительно, в качестве такого множества можно взять область применимости (относительно соответствующего алфавита) любого алгорифма с неразрешимой частной проблемой применимости.

Комментируя этот результат, можно заметить, что нормальный алгорифм с неразрешимой частной проблемой применимости строится как универсальный. Но на вход универсального алгорифма подается «объектный» алгорифм, запись которого

может быть сколь угодно сложной. Сложность этого конструктивного объекта не ограничена сверху, возникает как бы случайная $(0,1)$ -последовательность [10]. Естественно поэтому, что проблема применимости оказывается в этом случае неразрешимой. Но формальное доказательство основано на противоречии, обнаруживаемом в доказательстве леммы 1.

Заключение

В статье рассмотрена целостная методика преподавания теории алгоритмов на примере нормальных алгорифмов Маркова. Методика ориентирована на студентов, обучающихся по направлениям, связанным с разработкой программных технологий.

Новизна методики состоит в подходе к изложению самой концепции алгоритма, теоретико-множественном подходе к нормальным алгорифмам Маркова, в подборе некоторых примеров и приемов доказательства, где расставлены необходимые акценты; сравнение алгоритмических и игровых процессов (алгоритма и доказательства, в частности). Всё в целом ориентировано на студентов-программистов. С этим же связаны и определенные программистские аналогии (например, сравнение универсального алгорифма с программой-интерпретатором).

Основная цель состояла в изложении доступного для понимания студентов-нематематиков доказательства теоремы о неразрешимости проблемы применимости (останова) для нормальных алгорифмов.

Работа основана на личном опыте автора в преподавании теории алгоритмов. В качестве дальнейших методических задач можно назвать проработку курса математической логики для студентов тех же специальностей. Содержание статьи будет полезным преподавателям и студентам при подготовке к занятиям.

ССЫЛКИ НА ИСТОЧНИКИ

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 3-е изд.: Пер. с англ.- СПб.: ООО «Диалектика», 2019. – 1328 с.
2. Годлевский А.Б. О некоторых разрешимых и неразрешимых случаях проблемы функционально эквивалентности схем программ над памятью // Тезисы докладов III Всесоюзного симпозиума.- Изд. Кишиневского гос. ун-та, 1974. – С. 174-179
3. Подловченко Р.И. Исследования в теории алгебраических моделей программ с процедурами // Программирование. – 2016. - №1. – С. 5-9
4. Косовский Н.К., Косовская Т.М. Полиномиальный тезис Черча для РЕФАЛ-5-функций, нормальных алгоритмов и их обобщений // Компьютерные инструменты в образовании. – 2010. – С. 12-21
5. Косовский Н.К. Алгоритмы Маркова-Турчина и доказательства полиномиальной эффективности программ на языке РЕФАЛ-5 // Компьютерные инструменты в образовании.- 2012. - №4. – С. 41-49
6. Гурин Р.Ф., Романенко С.А. Язык программирования РЕФАЛ ПЛЮС / Учеб.пособие, Ун-т г. Переславля им. А.К. Айламазяна. – Переславль-Залесский, 2006
7. Марков А.А., Нагорный Н.М. Теория алгорифмов. – М.: Наука, 1984. – 432 с.
8. Кушнер Б.А. Лекции по конструктивному математическому анализу. – М.: Наука, 1973. – 448 с.
9. Белоусов А.И., Ткачев С.Б. Дискретная математика // Учеб. для вузов. – 5-е изд. – М.: Изд. МГТУ им. Н.Э. Баумана, 2015. – 744 с.
10. Хованский А.В. Индивидуальная случайная последовательность и генераторы случайных чисел // Математическое моделирование. – 2009. - №7. – С. 93-105.

Alexey Belousov

Candidate of Physical-Mathematical Sciences, Associate Professor, Bauman Moscow State Technical University, Moscow

al_belous@bk.ru

On the methodology for the presentation of some sections of the theory of algorithms: the applicability problem for normal Markov algorithms

Abstract. The relevance of the methodological problem under consideration is due to the fact that the theory of algorithms occupies a significant place in the curriculum of student programmers, and it requires the development of a rigorous and at the same time methodology for students who do not specialize in mathematics proper presentation of the foundations of this theory. The article proposes a methodology for presenting the key provisions of the theory of algorithms using the example of normal Markov algorithms. The algorithmic unsolvability of the applicability problem (or the stop problem) for normal algorithms is proved in detail. The main results necessary for understanding the main theorem are outlined. The content of the article will be useful to students of programming specialties, as well as teachers of courses in mathematical logic and theory of algorithms, courses in designing algorithms.

Keywords: algorithm, normal Markov algorithm, algorithmically unsolvable problems.

МЕТОД ЗОЛОТОГО СЕЧЕНИЯ И ДРЕВНЕРУССКИЕ САЖЕНИ

Аннотация

Актуальность исследуемой проблемы обусловлена необходимостью формирования и развития профессиональной культуры будущего инженера. Цель статьи состоит в том, чтобы подчеркнуть роль межпредметных связей в высшей технической школе как одного из факторов, формирующего профессиональные компетенции будущего специалиста, а также фактора, повышающего мотивацию обучения. В целом результаты опытно-экспериментальной работы позволяют говорить о состоятельности предлагаемого содержания для проведения семинарских занятий, дополнительной самостоятельной работы с целью профессионально-мотивирующего обучения студентов технических и математических специальностей. Содержание статьи будет интересным учителям, преподавателям математики, студентам, старшеклассникам.

Ключевые слова

метод золотого сечения минимизации функции, золотое сечение, межпредметные связи, древнерусские сажени

АВТОР

Вергазова Ольга Бухтияровна,
кандидат философских наук, доцент кафедры математического моделирования
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
olga.aika@yandex.ru

Введение

При изучении в курсе методов оптимизации метода золотого сечения, как одного из методов минимизации унимодальных функций, студенты обычно приводят традиционные примеры применения золотой пропорции (другие названия - золотое сечение, деление отрезка в среднем и крайнем отношении): древнеегипетская архитектура, шедевры древнегреческого искусства и искусства эпохи Возрождения. Изучением отражения золотой пропорции в искусстве, архитектуре, в объектах живой и неживой природы занимались как отечественные, так и зарубежные исследователи. Но интересные факты о связи золотой пропорции и древнерусской системы сажени,

как правило, остаются малоизвестными. Например, работы известных методистов математики, историков, специалистов в области истории математики и истории русской архитектуры, которые доказывают, что система древнерусских саженьей также построена на основе золотой пропорции.

В процессе изучения в курсе методов оптимизации метода золотого сечения с целью расширения кругозора студентов, формирования эстетического аспекта профессиональной культуры будущего инженера следует обязательно рассмотреть историю данного вопроса. Как правило, общеизвестные примеры золотой пропорции в природе и искусстве студентам знакомы, но для студентов технических и математических специальностей особый интерес представляют исследования специалистов в области истории математики и истории русской архитектуры о связи золотой пропорции и системы древнерусских саженьей.

Целью проведенной работы явилось изучение роли межпредметных связей в высшей технической школе как одного из факторов, формирующего профессиональные компетенции будущего специалиста, а также фактора, повышающего мотивацию и успешность обучения математике.

Методология и результаты исследования

Методы и способы проведения эксперимента были следующими. Были разработаны и проведены семинарские занятия по курсу методов одномерной оптимизации:

- с традиционным изложением материала (изучение алгоритма, решение задач на занятии, самостоятельное решение задач);

- сочетание традиционного изложения материала с привлечением дополнительных сведений о золотом сечении и его роли в системе древнерусских саженьей.

Во втором случае дополнительный материал излагался как преподавателем, так и группой студентов, подготовившими материал о золотой пропорции в природе и искусстве.

Основные сведения о знаменитой пропорции

Золотое сечение - это такое пропорциональное деление отрезка на неравные части, при котором весь отрезок так относится к большей части, как сама большая часть относится к меньшей. Пусть длина некоторого отрезка равна a , длина его большей части равна x , тогда $a - x$ - длина меньшей части отрезка. Составим отношение согласно приведенному выше определению:

$$(1) \frac{a}{x} = \frac{x}{a-x}.$$

В пропорции, как известно, произведение крайних членов равно произведению средних, поэтому от пропорции (1) перейдем к равенству

$$a(a - x) = x^2.$$

Отсюда получаем квадратное уравнение

$$x^2 + ax - a^2 = 0.$$

Корни уравнения являются иррациональными числами

$$x_1 = \frac{-a - a\sqrt{5}}{2}, \quad x_2 = \frac{-a + a\sqrt{5}}{2}.$$

Длина отрезка x выражается положительным числом, поэтому из двух корней следует выбрать положительный: $x = \frac{-a + a\sqrt{5}}{2}$ или $x = \frac{a(\sqrt{5}-1)}{2}$.

Число $x = \frac{a(\sqrt{5}-1)}{2}$ обозначается буквой Φ в честь древнегреческого скульптора Фидия (род. в начале V в. до н. э.), в творениях которого это число встречается многократно. Число Φ - иррациональное, с восемью десятичными знаками оно имеет вид: $\Phi = 0,61803398...$ Для практических целей часто используют приближенные значения:

0,618 (длина большей части отрезка) и 0,382 (длина меньшей части отрезка). Если длину отрезка принять за 100 частей, то большая часть отрезка равна 62, а меньшая - 38 частям.

Принято считать, что понятие о золотом сечении ввел в научный обиход Пифагор, который позаимствовал знание о нем у египтян и вавилонян во время своих путешествий. Платон в своем диалоге «Тимей» рассматривал математические и эстетические воззрения школы Пифагора, в частности, золотое сечение. [1-3]

Метод золотого сечения одномерной минимизации функции на отрезке

Рассмотрим метод минимизации функции на отрезке, при котором исключение части исходного отрезка происходит на основе золотой пропорции. В данном методе точки x_1 и x_2 расположены на отрезке $[a; b]$ так, что одна из них становится пробной точкой и на новом отрезке, полученном после исключения части исходного отрезка. Причем точки x_1 и x_2 обладают также следующим свойством: каждая из них делит отрезок на две неравные части так, что отношение длины всего отрезка к длине его большей части равно отношению длин большей и меньшей частей отрезка. То есть отрезок делится в золотом отношении, а точки x_1 и x_2 называются точками золотого сечения отрезка. (Рис. 1).

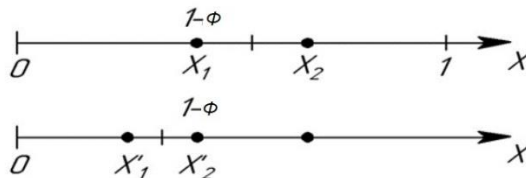


Рис. 1. К определению пробных точек в методе золотого сечения.

Для произвольного отрезка $[a; b]$ выражения для точек x_1 и x_2 имеют вид

$x_1 = \frac{-a - a\sqrt{5}}{2}$, $x_2 = \frac{-a + a\sqrt{5}}{2}$. В конце вычислений по данному методу в качестве приближенного значения x^* можно взять середину последнего и полученных отрезков $x^* \approx x_{\text{сред}} = \frac{a+b}{2}$.

Рассмотрим алгоритм метода золотого сечения.

ШАГ 1. Вычислить

$$x_1 = \frac{(a + (3 - \sqrt{5}))(b - a)}{2}, \quad x_2 = \frac{(a + (\sqrt{5} - 1))(b - a)}{2}.$$

Вычислить $f(x_1)$ и $f(x_2)$. Положить $\varphi = \frac{\sqrt{5}-1}{2}$, $\varepsilon_n = \frac{b-a}{2}$.

ШАГ 2. Проверка на окончание поиска: если $\varepsilon_n > \varepsilon$, то перейти к шагу 3, иначе - к шагу 4.

ШАГ 3. Переход к новому отрезку и новым пробным точкам.

Если $f(x_1) \leq f(x_2)$, то перейти к отрезку $[a; x_2]$, положив $b = x_2$, $x_2 = x_1$,

$f(x_2) = f(x_1)$, $x_1 = b - \varphi(b - a)$ и вычислить $f(x_1)$,

иначе - положить $a = x_1$, $x_1 = x_2$, $f(x_1) = f(x_2)$, $x_2 = a + \varphi(b - a)$ и вычислить $f(x_2)$.

Положить $\varepsilon_n = \varphi \cdot \varepsilon$ и перейти к шагу 2.

ШАГ 4. Окончание поиска: положить $x^* \approx x_{\text{сред}} = \frac{a+b}{2}$, $f^* \approx f(x_{\text{сред}})$.

Пример. Решить задачу одномерной минимизации функции

$f(x) = x^4 + e^{-x} \rightarrow \min$, где $0 \leq x \leq 1$, $\varepsilon = 0,1$, методом золотого сечения.

Итерация 1.

ШАГ 1. $x_1 = 0,382$, $x_2 = 0,618$. $f(x_1) = 0,704$, и $f(x_2) = 0,685$, $\varepsilon_n = 0,5$.

ШАГ 2. $\varepsilon_n = 0,5 > \varepsilon = 0,1$, поэтому переходим к шагу 3.

ШАГ 3. $f(x_1) > f(x_2)$, поэтому полагаем $a = x_1 = 0,382$, $x_1 = 0,618$,

$f(x_1) = 0,685$, $x_2 = 0,764$, $\varepsilon_n = 0,309$ и вычисляем $f(x_2) = 0,807$. Переходим к следующей итерации, начиная с шага 2.

Дальнейшие результаты запишем в таблицу 1.

Таблица 1

Номер итерации	a	b	ε_n	x_1	x_2	$f(x_1)$	$f(x_2)$	Сравнение $f(x_1)$ и $f(x_2)$
2	0,382	1,000	0,309	0,618	0,764	0,685	0,807	$f(x_1) < f(x_2)$
3	0,382	0,764	0,191	0,528	0,618	0,668	0,685	$f(x_1) < f(x_2)$
4	0,382	0,618	0,118	0,472	0,528	0,673	0,668	$f(x_1) > f(x_2)$
5	0,472	0,618	0,073	0,073				

$0,073 < 0,1$ - точность достигнута.

Таким образом, $x^* \approx \frac{0,472+0,618}{2} \approx 0,55$, $f^* \approx 0,67$. [4]

Золотая пропорция и система древнерусских сажень

В древнерусской числовой системе архитектурного пропорционирования, которая находила применение еще до монгольского нашествия, в качестве единиц измерения использовались сажени. Сажень было несколько, причем они были разной длины и несоизмеримы друг другу, при этом использовались при замере объектов одновременно. Точное количество видов сажень неизвестно, но историки и архитекторы признают, что существовали не менее семи типоразмеров сажень с собственными названиями.

Различные авторы по-разному определяют время возникновения древнерусской системы измерительных инструментов. Анатолий Федорович Черняев, инженер, академик Международной Академии информатизации, в своей работе «Золото Древней Руси. Русская матрица - основа золотых пропорций» ссылается на следующие две точки зрения.

Согласно первой, древнерусская система мер была заимствована у греческих колоний. А. Ф. Черняев, цитируя Беляева Г. Н., говорит, что система мер: «... занесена на русскую равнину, вероятно, задолго до утверждения там славян в III-II вв. до Р. Х. из Пергама через малоазиатские греческие колонии». [5]

Вторая точка зрения была отражена в работах историка с мировым именем Бориса Александровича Рыбакова. Он полагал, что большая часть мер появилась у славян в период XII-XIII вв. и развивалась, совершенствовалась примерно до XVII в. Но Б. А. Рыбаков также не исключал возможность привнесения в древнерусскую систему измерительных инструментов каких-либо элементов из других сопредельных и отдаленных стран. Отметим, что между указанными выше двумя крайними наметками времени почти полтора тысячелетия. [6, 7]

Точка зрения известного историка математики и педагога И. Я. Депмана (1885-1970) близка к точке зрения Б. А. Рыбакова. И. Я. Депман в своей «Истории арифметики» говорит следующее: «Первое упоминание о *сажени*, равной трем локтям, имеется в летописи под 1017. *Маховой саженью* называлось расстояние между концами пальцев распростертых рук, *косой саженью* - расстояние от каблука левой ноги до конца пальцев поднятой вверх правой руки (давалось и другое объяснение происхождения длины косой сажени)».

И. Я. Депман также обращается к вопросу о происхождении слова «сажень»: «В разных книгах название древнерусской меры «сажень» производили от английского слова fathom. Однако нет никакой надобности искать корень слова «сажень» (произносится: сАжень или сажЕнь) в иностранных языках. По «Толковому словарю живого великорусского языка» Владимира Даля слово «сажень» имело в старое время форму «сяжень». Глагол «сягать» сохранился в языке в форме «досягать», а также в словах «досягаемый», «недосягаемый», в выражениях: «разум сягает, да воля не владеет», «рука не сягает» и т. д. Отсюда естественное объяснение слова «сажень» или «сяжень»: то, что досягаемое рукой. [8]

Б. А. Рыбаков выделял две наиболее распространенные системы измерения: новгородско-псковскую и московско-владимирско-черниговскую. Он констатировал одновременное существование в Древней Руси с X по XVII вв. только 7 видов сажень и

локтей. По замечанию А. Ф. Черняева, Б. А. Рыбаков не отмечал в системе сажень такой единицы, как вершок.

Приведем таблицу длин сажень, выделенных Б. А. Рыбаковым, в которую дополнительно включим вершок. (Таблица 2).

Таблица 2

Сажени и доли сажень

Виды сажень	Сажень, см	Полсажени, см $\frac{1}{2}$ сажени	Локоть, см $\frac{1}{4}$ сажени	Пядь, см $\frac{1}{8}$ сажени	Пясть, см $\frac{1}{16}$ сажени	Вершок, см $\frac{1}{32}$ сажени
Простая	152,76	76,38	38,19	19,10	9,55	4,77
Маховая	176,40	88,20	44,10	22,05	11,03	5,51
Морская	183,00	91,50	45,75	22,88	11,44	5,72
Трубная	187,00	93,50	46,75	23,38	11,69	5,84
Без чети	197,20	98,60	49,30	24,65	12,32	6,16
Косая	216,00	108,00	54,00	27,00	13,50	6,75
Великая	249,46	124,73	62,37	31,18	15,59	7,80

Б. А. Рыбаков отметил следующие закономерности, определяющие соотношения между саженьями. Пусть $a = \frac{176,4}{2} = 88,2$ - половина маховой сажени. (Таблица 3).

Таблица 3

Соотношения между саженьями

Сажень	Длина, см
Простая (прямая)	$a\sqrt{3}=88,2\cdot 1,732=152,76$
Маховая (мерная)	$a\sqrt{4}=88,2\cdot 2=176,4$
Без чети (царская)	$a\sqrt{5}=88,2\cdot 2,236=197,21$
Косая (казенная)	$a\sqrt{6}=88,2\cdot 2,45=216,04$
Греческая	$a\sqrt{7}=88,2\cdot 2,646=233,4$
Великая	$a\sqrt{8}=88,2\cdot 2,828=249,46$

Зависимость $a\sqrt{7}$ добавлена Черняевым А. Ф., отметившим, что греческая сажень часто встречается при обмерах древних сооружений. А. Ф. Черняев отмечает, что Б. А. Рыбаков не рассматривал соотношения сажень по золотому сечению, а также не было объяснения, почему при измерении объектов существовала необходимость использования нескольких длин-сажень.

А. Ф. Черняев, на основе анализа работы архитектора А. А. Пилецкого, исследовавшего системы пропорционирования в древнерусской архитектуре, приводит набор 12 древних сажень, полученный методом усреднения многих образцов измерительных инструментов, который представлен в таблице 4. (Таблица 4).

Таблица 4

Виды сажень

Название сажени	Длина, см
Сажень городовая	284,8
Сажень без названия	258,4
Сажень великая	244,0
Сажень греческая	230,4
Сажень казенная	217,6
Сажень царская	197,4
Сажень церковная	186,4
Сажень народная	176,0
Сажень кладочная	159,7
Сажень простая	150,8
Сажень малая	142,4
Сажень без названия	134,5

Черняев А. Ф. отмечает следующую закономерность. Если найти последовательные отношения величин пяти самых больших сажень к пяти самым маленьким, то получим

$$\frac{284,8}{176} = \frac{258,4}{159,7} = \frac{244}{150,8} = \frac{230,4}{142,4} = \frac{217,6}{134,5} = 1,618.$$

Таким образом, отношение длин указанных видов сажень близко к золотому отношению. Для доказательства пропорциональности числу $\Phi=1,618$ оставшихся царской и церковной сажень достаточно удвоить длину кладочной и простой сажень и разделить полученные результаты на длину царской и церковной сажень: $\Phi=159,7 \cdot 2 / 197,4 = 150,8 \cdot 2 / 186,4 = 1,618$. По мнению А. Ф. Черняева, в структуру древнерусских сажень заложено пропорционирование по золотому сечению, то есть, кратность числу $\Phi=1,618033989...$

А. Ф. Черняев заключает, что соразмерность и эстетическая привлекательность памятников древнерусского зодчества являются именно следствием особой, подвижной функции взаимосвязанного комплекса древнерусских сажень, заключающейся в том, что их основное назначение - соизмерение. ***Сажени - инструмент соизмерения и пропорционирования.*** А. Ф. Черняев подчеркивает, что сажени «не статические линейки, а остановленные длиной продолжающиеся динамические процессы». Именно поэтому архитектурные памятники Древней Руси своей соразмерностью и пропорциональностью превосходят современные архитектурные сооружения, возведенные по точному стандартному метру. [9] Приведем слова исследователя А. И. Азевича: «Сопряженность русских мер была основой гармоничных решений в архитектурных сооружениях. Создав систему сажень, основанную на пропорциях человеческого тела, русские зодчие получили мощный инструмент для тонкого архитектурного варьирования, передачи в пропорциях целой гаммы оттенков». [10]

А. Ф. Черняев рассматривал также следующий способ восстановления эталонной сажени. Если взять деревянный прут длиной в рост плотника, например, 172 см, и принять его за базисную длину. Если три прутка данной длины сложить равносторонним треугольником, то его высота будет равна 148,96 см, что близко по длине сажени простой. Если к середине мерной сажени под прямым углом приставить другую мерную сажень и соединить их свободные концы длинными прутками, то получим равнобедренный треугольник с боковыми сторонами, равными 197,22 см - аналог «сажени без чети». Возьмем две полученные простые сажени, соединим их концы под прямым углом. В полученном прямоугольном равнобедренном треугольнике гипотенуза равна 216,04 см - аналог сажени косой. В прямоугольном равнобедренном треугольнике с катетами, равными мерной сажени, гипотенуза равна 243,24 - сажени великой. Наконец, трубную сажень можно получить, если к середине косой сажени под прямым углом приставляется сажень простая. При соединении их свободных концов получим равнобедренный треугольник с боковой стороной 187,08 см - аналог трубной сажени.

Восстановление основных сажень окончено. Не восстановлена только морская сажень. Но А. Ф. Черняев отмечает, что и Б. А. Рыбаков сомневался в существовании морской сажени как самостоятельной меры. [11]

Заключение

Использование дополнительного материала о золотой пропорции в процессе обучения математике формирует представление студента о том, что золотое сечение - один из основных принципов представления человека о красивом. Золотая пропорция, различные виды симметрии - причина эстетической привлекательности предметов искусства, объектов в живой и неживой природе и в технике.

В обоих случаях студенты выполняли домашнее задание, которое заключалось в выполнении минимизации одномерной функции несколькими методами. Всего из рассмотренных шести методов нужно было применить не менее четырех различных способов решения задачи. Анализ выполнения домашнего задания по данной теме показал следующее. В группах с традиционным изложением материала из 41 студента метод золотого сечения успешно применили 26 человек (63,4 %). В группах, в которых занятия были проведены с подробным рассмотрением истории вопроса о золотом сечении, из 37 человек успешно справились с решением задачи методом золотого 35 студентов (94,6 %).

Великому астроному XVI в. Иоганну Кеплеру принадлежат слова: «Геометрия владеет двумя сокровищами: одно из них - это теорема Пифагора, а другое деление отрезка в крайнем и среднем отношении... Первое можно сравнить с мерой золота, второе же больше напоминает драгоценный камень».

В заключение остается добавить, что золотая пропорция перестала быть сокровищем только геометрии. Исторические примеры, искусство и современная научная мысль показывают, что число $\Phi=1,618033989...$ - это универсальная мировая константа, знание которой необходимо школьникам и студентам. Знакомство студентов математических и технических специальностей с историческими и эстетическими аспектами математики является необходимой составляющей формирования и развития профессиональной культуры будущих инженера, математика или преподавателя математики.

ССЫЛКИ НА ИСТОЧНИКИ

1. Романенко Ю. М. Эстетика и математика. М. Издательство МГОУ, 2005. – 108 с.
2. Корбаллан Ф. Золотое сечение. Математический язык красоты/ пер. с англ. – М.: Де Агостини, 2014. – 160 с.
3. Тимердинг Генрих Е. Золотое сечение. пер. с нем./ под. ред. Г. М. Фихтенгольца. Изд. 2-е, стереотипное. М.: Комкнига, 2005. 88 с.
4. Лесин В. В., Лисовец Ю.П. Основы методов оптимизации. – М.: Изд-во МАИ, 1995. – 344 с.
5. Черняев А. Ф. Золото Древней Руси. Русская матрица – основа золотых пропорций. М. РусАрх.1998. URL: <http://www.rusarch.ru/chernyaev.htm>
6. Рыбаков Б. А. Рождение Руси. М.: «АиФ Принт», 2004. 447 с.
7. Рыбаков Б. А. Русские системы мер длины XI-XV веков// Советская этнография. – 1949. 1. С. 67-91.
8. Депман И. Я. История арифметики. М.: Комкнига. 2006. 416 с.
9. Черняев А. Ф. Указ. соч.
10. Азевич А. И. Двадцать уроков гармонии: Гуманитарно-математический курс. – М.: Школа-Пресс, 1998. – 160 с.: ил. (Библиотека журнала «Математика в школе». Вып. 7).
11. Черняев А. Ф. Указ. соч.

Olga B. Vergazova

Candidate of Philosophy, Associate Professor, Department of Mathematical modeling, Bauman Moscow State Technical University, Moscow

olga.aika@yandex.ru

The method of the Golden section and ancient Russian fathoms

Annotation. The relevance of the problem is due to the need for the formation and development of professional culture of the future engineer. The purpose of the article is to emphasize the role of interdisciplinary connections in higher technical school as one of the factors that forms the professional competence of the future specialist, as well as a factor that increases the motivation of learning. In General, the results of experimental work allow us to talk about the consistency of the proposed content for seminars, additional independent work for the purpose of professional and motivating training of students of technical and mathematical specialties. The content of the article will be interesting for teachers, teachers of mathematics, students, high school students.

Key words: Golden section method of function minimization, Golden section, intersubject connections, old Russian fathoms.

К ЗАДАЧЕ О ФОРМЕ ЯЧЕЕК СОЛНЕЧНЫХ БАТАРЕЙ

Аннотация

Актуальность исследуемой проблемы обусловлена необходимостью создания надежных и эффективных панелей солнечных батарей. В целом результаты изучения теоретических источников и опытно-экспериментальной работы позволяют говорить о состоятельности предлагаемого содержания в решении проблемы выбора формы элементарной ячейки солнечной батареи. Актуальность исследуемой проблемы обусловлена также и необходимостью формирования и развития профессиональной культуры будущего инженера. Содержание статьи будет интересным преподавателям, студентам, старшеклассникам.

Ключевые слова

панель солнечной батареи, элементарные ячейки, правильные многоугольники

АВТОРЫ

Вергазова Ольга Бухтияровна,

кандидат философских наук, доцент кафедры математического моделирования
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
olga.aika@yandex.ru

Кваша Владимир Сергеевич,

кандидат технических наук, старший преподаватель отдела №1 учебного военного центра
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
Derwolf-82@mail.ru

Федоров Никита Николаевич,

студент кафедры космических приборов и систем
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
fedorovnikita235@gmail.com

Бекеров Ильмир Дамирович,

студент кафедры прикладной механики и управления
ФГБОУ ВО «Московский государственный университет
им. М. В. Ломоносова», г. Москва
omert14@yandex.ru

Введение

Солнечные батареи активно используются как на земле, так и в космосе. В космосе они играют важное значение, так как обеспечивают питание искусственных спутников Земли солнечной энергией без других внешних источников энергообеспечения. Современный мир невозможно представить без спутниковой связи, поэтому создание долговечных и надежных солнечных батарей является актуальной и важной задачей.

Методология и результаты исследования

Панель солнечной батареи представляет собой набор элементарных ячеек (Рис. 1).

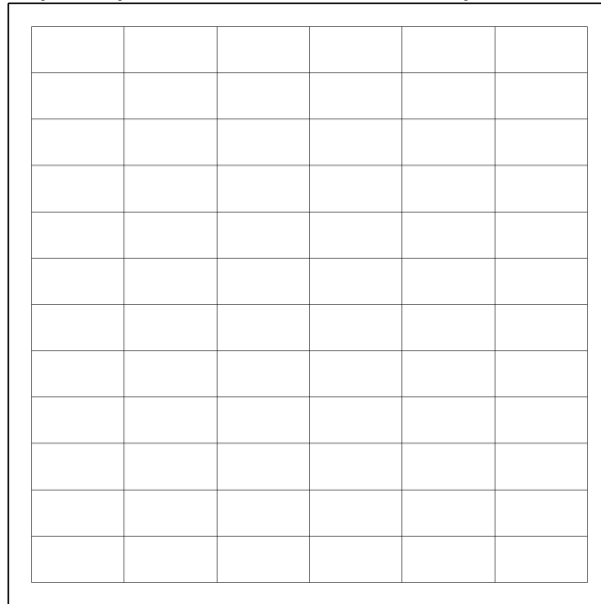


Рис.1. Общий вид панели солнечной батареи.

Проанализируем, насколько зависит работоспособность и эффективность батареи от того или иного вида конфигурации ячеек, и как эти характеристики зависят от формы ячеек. Рассмотрим несколько конфигураций разбиения полотна на ячейки в форме правильных многоугольников, которые можно применить в случае замощения плоскости: треугольник, квадрат и шестиугольник.

Сравним периметры равновеликих (имеющих равные площади) правильных треугольника, четырехугольника (квадрата) и шестиугольника.

Пусть: s - площадь контакта метеорита, попавшего в солнечную батарею. Запишем функцию $f = p(s)$, выражающую зависимость площади одной элементарной ячейки от ее периметра.

Пусть элементами ячейки являются:

- 1) *квадрат со стороной a_4* : $s = a_4^2$, $a_4 = \sqrt{s}$, $p_4 = 4\sqrt{s}$,
- 2) *правильный треугольник со стороной a_3* : $s = \frac{a_3^2\sqrt{3}}{4}$, $a_3 = \frac{4\sqrt{s}}{\sqrt{3}}$, $p_3 = \frac{6\sqrt{s}}{\sqrt{3}}$,
- 3) *правильный шестиугольник со стороной a_6* : $s = \frac{3a_6^2\sqrt{3}}{2}$, $a_6 = \frac{\sqrt{2s}}{\sqrt{3}\sqrt{3}}$,
 $p_6 = \frac{6\sqrt{s}\sqrt{2}}{\sqrt{3}\sqrt{3}}$.

Таким образом, имеем: $p_4 = 4\sqrt{s}$, $p_3 = \frac{6}{\sqrt{3}}\sqrt{s}$, $p_6 = \frac{6\sqrt{2}}{\sqrt{3}\sqrt{3}}\sqrt{s}$.

Сравнивая полученные коэффициенты, получим, что p_6 - наименьшая величина, а p_3 - наибольшая. Таким образом, далее не будем рассматривать вариант с треугольными ячейками как нерациональный.

Теперь представим, что наша панель - это некоторая область площади s_1 , которую разбивают на ячейки площадью s_0 так, что $\sum s_0 = S_1$. Будем считать, что ячейка выведена из строя, если в нее попадает метеоритный камень. Предположим, что площадь контакта метеоритного камня - это площадь, окружности радиуса r . Тогда выясним, как будет меняться количество выбитых ячеек в зависимости от их формы. Рассмотрим дискретную случайную величину X - количество выбитых прямоугольных

ячеек. В таком случае вероятность выбивания i ячеек найдем как $P_i = \frac{\Delta s_0^i}{s_0}$, где Δs_0^i - область ячейки, попадание в которую соответствует выбиванию i числа ячеек.

Очевидно, что нет необходимости рассматривать всю площадь S_1 панели солнечной батареи, достаточно выяснить, какой области на одной элементарной ячейке Δs_0^i соответствует выбивание i числа ячеек.

Если мы разобьем плоскость S_1 на равные прямоугольники, как это реализовано в уже имеющихся конструкциях, солнечных батарей, площади Δs_0^i , то с.в. X будет принимать 4 значения:

$$X = \begin{cases} 1, \text{ если выбита 1 ячейка} \\ 2, \text{ если выбита 2 ячейки} \\ 3, \text{ если выбита 3 ячейки} \\ 4, \text{ если выбита 4 ячейки} \end{cases}$$

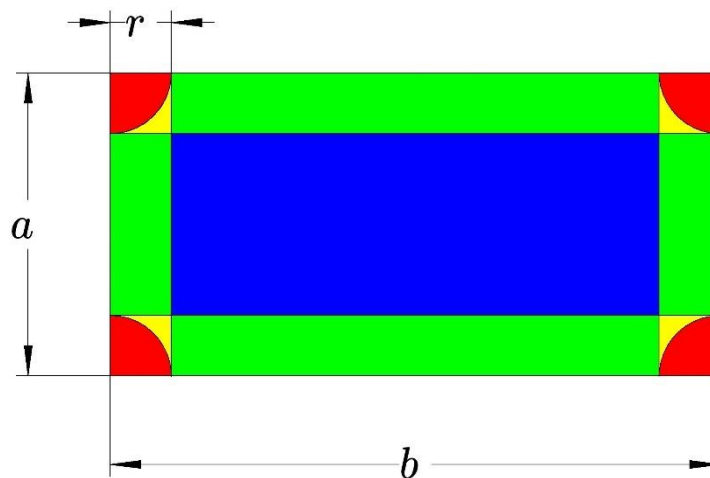


Рис. 2. Разбиение плоскости на равные прямоугольники.

На рис. 2 представлена прямоугольная ячейка, на которой красным цветом показана область, соответствующая выбиванию 4 ячеек, желтым - 3 ячеек, зеленым - 2 ячеек, синим - 1 ячейки, где a, b, r - высота, длина прямоугольной ячейки, радиус окружности, соответствующий площади контакта метеоритного камня с полотном солнечной батареи. (Рис. 2).

Тогда вероятность P_i , которая соответствует выбиванию i числа ячеек:

$$P_1 = \frac{(a-2r)(b-2r)}{ab}, \text{ если будет выбита одна ячейка,}$$

$$P_2 = \frac{2r(b+a-4r)}{ab}, \text{ если будут выбиты две ячейки,}$$

$$P_3 = \frac{4(r^2 - \frac{1}{4}\pi r^2)}{ab}, \text{ если будут выбиты три ячейки,}$$

$$P_4 = \frac{\pi r^2}{ab}, \text{ если будут выбиты четыре ячейки.}$$

Легко убедиться, что

$$\sum_{i=1}^4 P_i = 1.$$

$$P\left(|X_{\text{ср}} - M_X| < \frac{1}{n}\right) \xrightarrow{n \rightarrow \infty} 0, \text{ где } n - \text{число камней, попавших в солнечную батарею.}$$

Таким образом, найдем M_X - математическое ожидание дискретной случайной величины X :

$$M_X = \sum_{i=1}^4 P_i X_i = 1 + \frac{2r(a+b) + \pi r^2}{ab}$$

Проделаем аналогичный расчет для шестиугольной ячейки. На рис. 3 представлена шестиугольная ячейка, на которой желтым цветом показана область, соответствующая выбиванию трех ячеек, зеленым - двух ячеек, синим - одной ячейки. (Рис. 3).

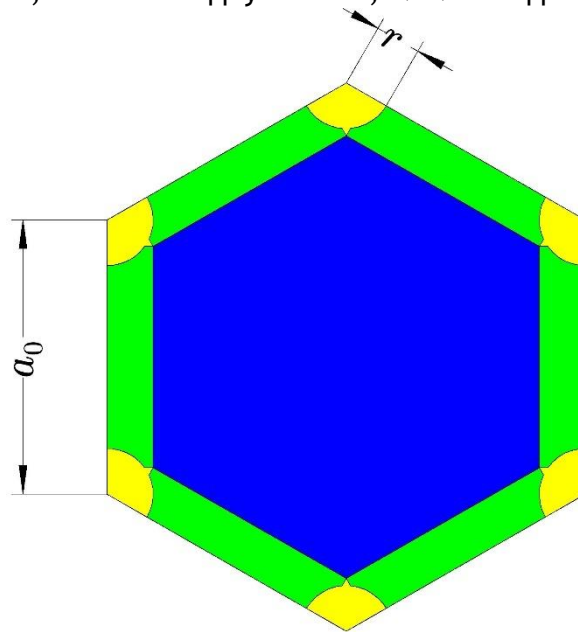


Рис. 3. Разбиение плоскости на правильные шестиугольники.

$$P_1 = \left(1 - \frac{2r}{\sqrt{3}a_0}\right)^2, \text{ если будет выбита одна ячейка;}$$

$$P_2 = \frac{4r}{\sqrt{3}a_0} - \frac{2r^2}{3a_0^2} - \frac{2\pi r^2}{3\sqrt{3}a_0^2}, \text{ если будут выбиты две ячейки;}$$

$$P_3 = \frac{12 + 2\sqrt{3}\pi r^2}{9a_0^2}, \text{ если будут выбиты три ячейки.}$$

$$Y = \begin{cases} 1, \text{ если выбита 1 ячейка} \\ 2, \text{ если выбита 2 ячейки.} \\ 3, \text{ если выбита 3 ячейки} \end{cases}$$

Рассмотрим дискретную случайную величину X - количество выбитых шестиугольных ячеек.

$$M_Y = \sum_{i=1}^3 P_i Y_i = 1 + \frac{36r^2 + 2\sqrt{3}\pi r^2}{9a_0^2} + \frac{4\sqrt{3}r^2}{3a_0}.$$

Сравним значения M_X и M_Y . Для этого рассмотрим равновеликие фигуры, площадь которых равна площади прямоугольника со сторонами $a \times b$. (Рис. 1).

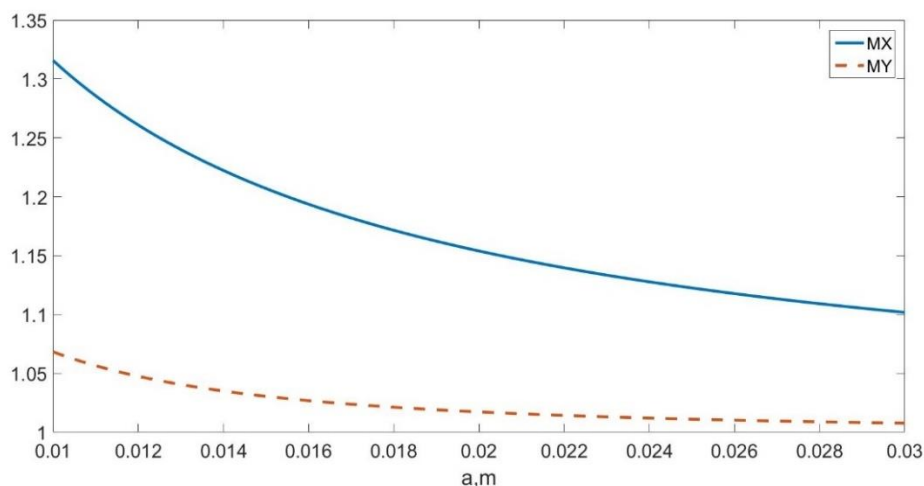
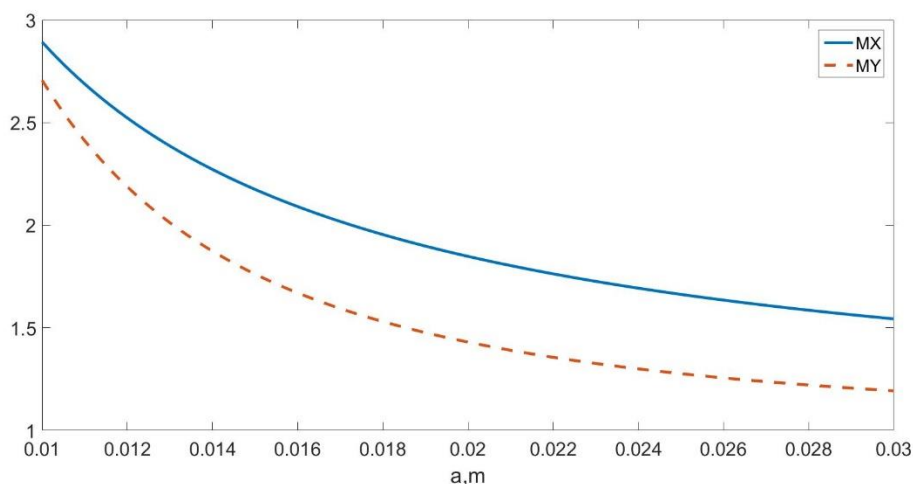
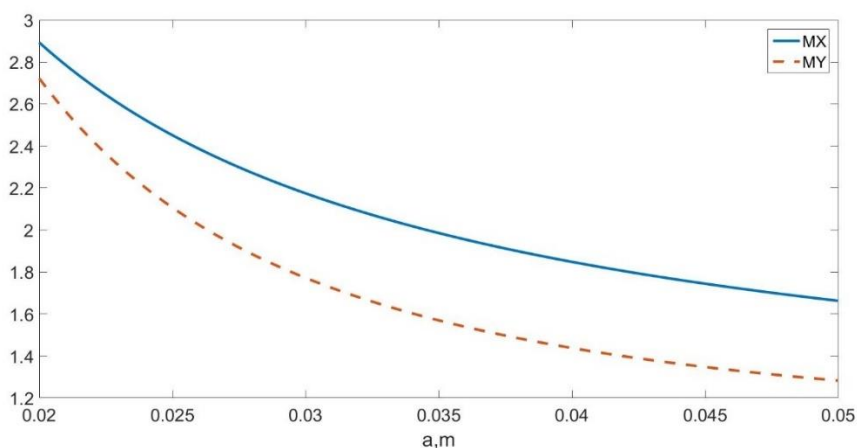
Пусть $\frac{b}{a} = k$, тогда $b = ak$, $S_{\text{равновелик.}} = a(ak) = ka^2$, где $S_{\text{равновелик.}}$ - заданная площадь прямоугольника, k - константа.

Выразим сторону шестиугольника площади $S_{\text{равновелик.}}$, воспользовавшись ранее полученной формулой $a_6 = \frac{\sqrt{2s}}{\sqrt[4]{3\sqrt{3}}}$. $a_0 = \frac{a\sqrt{2k}}{\sqrt[4]{3\sqrt{3}}}$. (Рис. 3).

После преобразований M_X и M_Y будут выглядеть следующим образом.

$$M_X = 1 + \frac{2ra(1+k) + \pi r^2}{a^2 k}, \quad M_Y = 1 + \frac{r^2(36 + 2\sqrt{3}\pi) + \pi r^2}{2\sqrt{3}ka^2} + \frac{12r^2}{a\sqrt{2k}}. \quad [1-8]$$

На рисунках 4, 5, 6 представлены графики, где показаны $M_X(a)$ и $M_Y(a)$ при заданных значениях r и k .

Рис 4. $r = 0.001m, k = 2$.Рис. 5. $r = 0.005m, k = 2$.Рис. 6. $r = 0.01m, k = 2$.

Таким образом, мы видим, что $M_X(a)$ и $M_Y(a)$ представляют из себя две убывающие функции, причем $M_Y(a)$ убывает быстрее, чем $M_X(a)$. Из графиков так же можно заключить, что при одинаковых значениях a , то есть и при одинаковых значениях площадей ячеек разной формы, M_X лежит выше M_Y , что обозначает, что будет выбито большее число ячеек прямоугольной формы в сравнении с шестиугольной ячейкой той же площади. Значения параметров r и k выбраны из соображений того, что если радиус окружности, являющейся площадью контакта метеоритного камня, r не будет

мал, в сравнении с габаритами элементарной ячейки и самой панели, то произойдет конструктивная гибель всей солнечной батареи. Коэффициент $k = 2$ выбран из конструктивных соображений, так как прямоугольные ячейки с $\frac{b}{a} \approx 2$ применяются в настоящее время. [9, 10]

Заключение

Таким образом, рассуждения на основе элементарной математики и основных понятий теории вероятностей обосновывают простой принцип конструирования панелей солнечных батарей - необходимо использовать правильные многоугольники (с целью замощения всей плоскости). При этом, при выборе из равновеликих многоугольников предпочтительнее фигуры с меньшим периметром.

ССЫЛКИ НА ИСТОЧНИКИ

1. Боровков А. А. Теория вероятностей. – М.: Наука, 1976.
2. Гнеденко Б. В. Курс теории вероятностей. – 5-е изд. – 1979.
3. Кендалл М., Моран П. Геометрические вероятности/ Пер. с англ. – М.: Мир, 1972.
4. Коваленко И. Н., Филиппова А. А. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1973.
5. Колчин В. Ф., Севастьянов Б. А., Чистяков В. П. Случайные размещения. – М.: Наука, 1976.
6. Колмогоров А. Н. Основные понятия теории вероятностей. – М.: Наука, 1974.
7. Прохоров Ю. В., Розанов Ю. А. Теория вероятностей (Основные понятия, предельные теоремы, случайные процессы). – 2-е изд. – М.: Наука, 1973.
8. Смирнов Н. В., Дунин-Барковский И. В. Курс теории вероятностей математической статистики для технических приложений. – 3-е изд. – М.: Наука, 1969.
9. Усков А.Е., Дайбова Л.А., Гиркин А.С. и др. Солнечные фотоэлектрические станции: перспективы, особенности работы и расчета экономической эффективности. – Политематический сетевой электронный журнал Кубанского государственного аграрного университета. 2015.
10. Соколов Е. В., Пенджола Р.И. Методы оценки стоимости создания солнечных батарей пилотируемых космических аппаратов на ранних этапах разработки изделий. – Экономика и управление: проблемы, решения. 4 (5-2). 2017. С. 263-266.

Vergasova Olga Bukhtiyarova,

Candidate of philosophy, associate Professor, Department of mathematical modeling, Moscow state technical University. N. E. Bauman", Moscow

olga.aika@yandex.ru

Kvasha Vladimir Sergeevich,

Candidate of technical Sciences, senior lecturer, Department No. 1, military training center, Moscow state technical University. N. E. Bauman", Moscow

Derwolf-82@mail.ru

Nikita Fedorov,

Student of the Department of space instruments and systems, Moscow state technical University. N. E. Bauman", Moscow,

fedorovnikita235@gmail.com

Bekirov Ilmir Damirovich,

Student of the Department of applied mechanics and control, MOSCOW state University. Lomonosov", Moscow,

omert14@yandex.ru

On the problem of the shape of solar cells

The urgency of the problem is due to the need to create reliable and efficient solar panels. In General, the results of the study of theoretical sources and experimental work allow us to speak about the consistency of the proposed content in solving the problem of choosing the form of a solar cell. The relevance of the problem is also due to the need for the formation and development of professional culture of the future engineer. The content of the article will be interesting to teachers, students, high school students.

Keywords: solar panel, elementary cells, regular polygons.

ЭЛЕМЕНТАРНОЕ ВВЕДЕНИЕ В ТЕОРИЮ ОДНОМЕРНЫХ ДИСКРЕТНЫХ ДИНАМИЧЕСКИХ СИСТЕМ

Аннотация

В статье на элементарном уровне изложено введение в теорию динамических систем на примере их простейших представителей - одномерных дискретных отображений. Рассматриваются основные понятия, возникающие при исследовании таких систем: неподвижные точки, их устойчивость, орбиты, детерминированность, хаос и др. Предназначается школьникам старших классов и студентам первого курса для первоначального самостоятельного знакомства с новой бурно развивающейся математической дисциплиной - хаотической динамикой.

Ключевые слова

одномерное отображение, траектория, неподвижная точка, хаос

АВТОРЫ

Грибов Александр Федорович,
кандидат физико-математических наук, доцент
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
alexandr-gribov@list.ru

Краснов Игорь Константинович,
кандидат технических наук, доцент
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
igorkrsnv@yandex.ru

Введение

Развитие нелинейной динамики привело к осознанию того, что простые нелинейные динамические системы могут приводить к решениям, поведение которых невозможно предсказать уже на конечных интервалах времени. Под динамическими системами понимают системы, состояние которых изменяется во времени. Математическими формами таких систем являются различные дифференциальные уравнения, а также отображения. Первым кто заметил, что в некоторых механических системах могут быть хаотические движения, был великий французский ученый Анри Пуанкаре. Однако в 1892 году это не было воспринято с должным вниманием и только после публикации результатов исследования метеоролога Е. Лоренца ситуация существенно изменилась. Были развиты методы исследования нелинейных систем, позволившие анализировать явления, которые ранее казались неподдающимися описанию.

Изобретение высокоскоростных компьютеров в 1950 годах стало переломным в истории динамики, и привело к многочисленным открытиям. Стало ясно, что явление хаоса чрезвычайно часто встречается в природе, а не является чем-то исключительным.

Поведение системы со сложной динамикой чрезвычайно зависимо от исходных данных, и небольшие изменения в них могут привести к непредсказуемым послед-

ствиям. Это означает, что если взять два близких начальных значения, то через конечное время (конечное количество итераций - в случае дискретной системы) траектории из этих точек будут значительно отличаться. Это явление Эдвард Лоренц назвал "Эффектом бабочки" в своей статье "Предсказание: взмах крыльев бабочки в Бразилии вызовет торнадо в штате Техас" в 1972 году. Под взмахом крыльев подразумеваются небольшие изменения в начальных условиях, которые могут привести к непредсказуемым последствиям. "Эффект бабочки" вызывает аллюзию к рассказу фантаста Р. Брэдли - "И грянул гром", где гибель бабочки в прошлом вызывает сильные изменения в окружающей среде мира в будущем.

Возникновение хаоса в динамических детерминированных системах кажется невозможным, так как система подчиняется определенному закону, и, зная начальные данные можно без труда предсказать конечное состояние системы. На самом деле противоречия нет. Великий А. Пуанкаре писал: «...совершенно ничтожная причина, ускользающая от нас по своей малости, вызывает значительное действие, которое мы не можем предусмотреть... Предсказание становится невозможным, мы имеем перед собой явление случайное» 1908г. «Наука и метод».

Биолог Роберт Мэй, изучив сложное поведение логистического отображения, опубликовал свою статью, в конце которой он пришел к такому выводу: "Вероятно, для всех нас было бы гораздо лучше, если бы не только при обучении или в научной работе, но и в повседневной политической и экономической жизни как можно больше людей поняло, что простые динамические системы не обязательно приводят к простому поведению".

Основной целью пособия является элементарное, насколько это возможно, изложение начальных понятий теории одномерных дискретных отображений, которое должно быть понятным студентам первого курса. Мы полагаем, что знакомство с этим пособием подтолкнет студентов к более детальному изучению этого быстро развивающегося раздела современной математики и нелинейной физики. Учитывая предназначение пособия, в нем часто строгие математические определения и доказательства заменяются упрощенными рассуждениями, приводящими, однако, к верным результатам. Большая часть исследований проводится с помощью компьютера и, поэтому, читателю целесообразно попытаться написать соответствующие программы для проведения самостоятельных исследований.

Одномерное отображение

Одномерная дискретная динамическая система (одномерное отображение) определяется последовательностью $x_{n+1} = f(x_n)$, $n=0,1,\dots$, где $f(x)$ - скалярная функция. Траекторией или орбитой называют последовательность $x_0, f(x_0), f(f(x_0)), f(f(f(x_0))), \dots$

Будем обозначать композицию функций с собой, как $f^2(x) = f(f(x))$; $f^3(x) = f(f(f(x)))$. Такую зависимость можно изобразить графически с помощью диаграммы Кёнигса-Ламерея [1], изображенной на рис.1.

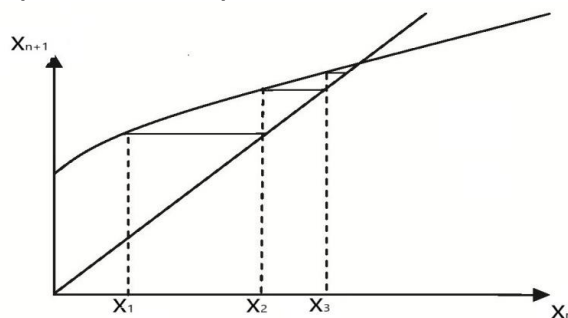


Рис. 1. Диаграмма Ламерея

Для этого к графику добавим биссектрису $y=x$ и построим последовательность по следующему правилу: задав значение x_1 , по графику находим $x_2 = f(x_1)$, данное значение $f(x_2)$ переносится на биссектрису - получаем значение $x_3 = f(x_2)$ и так далее.

Пусть x^* удовлетворяет условию $f(x^*) = x^*$. Тогда x^* есть неподвижная точка. Геометрически это точка пересечения графика $y=f(x)$ с биссектрисой $y=x$. Если $x_n = x^*$, то $x_{n+1} = f(x_n) = f(x^*) = x^*$. То есть орбита не покидает x^* для всех будущих итераций. Неподвижную точку называют устойчивой (притягивающей), если существует такая окрестность U этой точки, что $fU \subset U$ и $\lim_{n \rightarrow \infty} f^n(x) = x^* \forall x \in U$.

Для определения устойчивости x^* мы рассмотрим близкую орбиту $x_n = x^* + \delta_n$ и выясним, когда орбита притягивается или отталкивается от x^* , т.е. отклонение δ_n растет или уменьшается с ростом n .

$$x_{n+1} = x^* + \delta_{n+1} = f(x^* + \delta_n) = f(x^*) + f'(x^*)\delta_n + o(\delta_n)$$

Ограничиваясь линейным членом и учитывая, что $f(x^*) = x^*$ получаем линеаризованное отображение $\delta_{n+1} = f'(x^*)\delta_n$, в котором $\lambda = f'(x^*)$ называют мультипликатором. Запишем несколько итераций: $\delta_1 = \lambda\delta_0$, $\delta_2 = \lambda\delta_1 = \lambda^2\delta_0$, ..., $\delta_n = \lambda^n\delta_0$. Если $|\lambda| = |f'(x^*)| < 1$, то $\delta_n \rightarrow 0$ при $n \rightarrow \infty$ и неподвижная точка является устойчивой и наоборот, если $|f'(x^*)| > 1$ неподвижная точка является неустойчивой. В случае, когда $|f'(x^*)| = 1$, условия локальной стабильности определяются отброшенным членом.

Точка x и орбита $O(x)$ называются периодическими с периодом k , если $f^k(x)=x$, $f^i(x) \neq x$ для любого $i=1,2,...,k-1$. Говорят, что в этом случае отображение имеет k -цикл. Точки цикла $x_1, ..., x_k$ называют k кратными неподвижными точками. Например, периодическая орбита периода два состоит из двух точек x_1 и x_2 ; $x_2 = f(x_1)$, $x_1 = f(x_2)$; $x_1 = f^2(x_1)$, $x_2 = f^2(x_2)$. Цикл периода три состоит из трех точек x_1, x_2 и x_3 . $x_2 = f(x_1)$, $x_3 = f(x_2) = f^2(x_1)$, $x_1 = f(x_3) = f^2(x_2) = f^3(x_1)$. Периодическая орбита периода k состоит из k точек. Отметим, что k -кратная неподвижная точка является однократной неподвижной точкой для $f^k(x)$. Для k кратной неподвижной точки условия устойчивости и неустойчивости имеют вид $|f'(x_1) \dots f'(x_k)| < 1$ и $|f'(x_1) \dots f'(x_k)| > 1$.

Наиболее просто такие отображения выглядят в случае взаимно однозначных отображений, то есть

- 1) любому x_n отвечает единственное $x_{n+1} = f(x_n)$,
- 2) для любых x_n, y_n : $x_n \neq y_n \Rightarrow f(x_n) \neq f(y_n)$.

Следовательно, либо $f'(x) \geq 0$, либо $f'(x) \leq 0$. Тогда для взаимно однозначного отображения с $f'(x) \geq 0$ вся прямая в простейшем случае разбивается на некоторое число областей притяжения устойчивых неподвижных точек рис. 2.

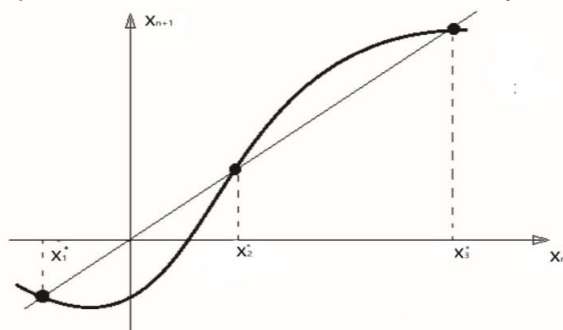


Рис. 2 Взаимно однозначное отображение для возрастающей функции

x_1^*, x_3^* - устойчивые неподвижные точки; x_2^* - неустойчивая неподвижная точка.

В случае $f'(x) \leq 0$ график функции пересекается с биссектрисой только в одной точке (устойчивой или неустойчивой). График такой функции показан на рис. 3. При

этом отображение $f^2(x)$ будет уже с производной больше нуля, т.е. это будет отображение первого типа. Неподвижной точке x_1^* отображения $f^2(x)$ соответствует двукратный цикл точек x_1^* и $x_2^* = f(x_1^*) \neq x_1^*$. В этом случае вся прямая разбивается на области притяжения двукратных неподвижных точек и область притяжения однократной неподвижной точки в случае ее устойчивости.

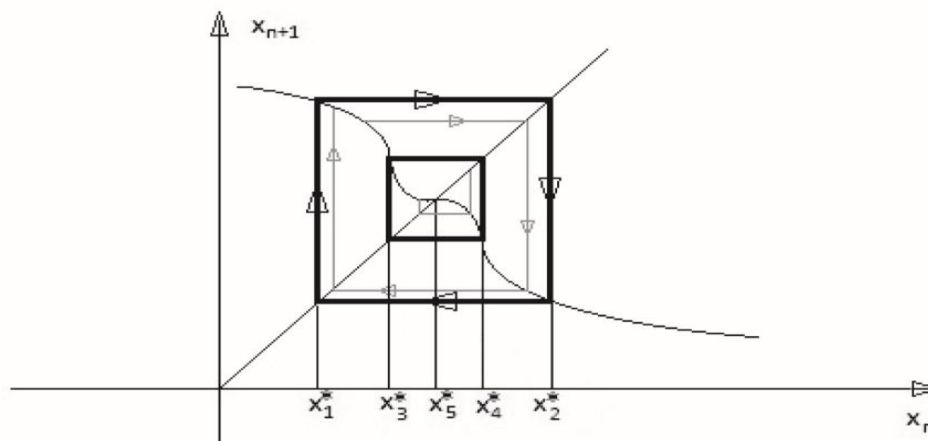


Рис. 3. Взаимно однозначное отображение для убывающей функции

Отображения «Зуб Пилы» и «Тент»

Существенно более сложная картина наблюдается при рассмотрении однозначного, но не взаимно однозначного отображения [2]. Здесь возможно появление очень сложных структур. Так появляются последовательности, которые начинаются с первоначально близких точек и существенно расходятся за конечное число итераций. Это свойство называют чувствительностью к заданию начальных условий. Отображение «зуб пилы» задается соотношением: $x_{n+1} = 2x_n \pmod{1}, x \in [0; 1]$. График такой функции представлен на рис. 4.

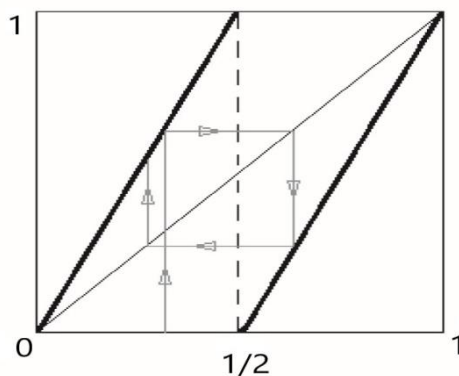


Рис. 4. Отображение «зуб пилы» - сдвиг Бернулли

Здесь первая половина интервала $[0, \frac{1}{2}]$ отображается на весь интервал $[0, 1]$. Вторая половина отображается тоже на весь интервал. Таким образом, если представить, что точка пробегает весь интервал $[0, 1]$, ее образ проходит этот интервал дважды. Растяжение приводит к чувствительности к заданию начальных данных. Наложение же образов при однократном проходе интервала, приводит к «перемешиванию» образов различных точек и не позволяет однозначно указать прообраз точки. Рассмотрим для такого отображения две последовательности с разными близкими начальными данными с $x_0 = 0,24$ и $x_0 = 0,235$.

0,240; 0,48; 0,96; 0,92; 0,84; 0,68; 0,36

0,235; 0,47; 0,94; 0,88; 0,76; 0,52; 0,04

Из последовательности видно, что незначительные различия в начальных данных приводят к существенно различным траекториям. Запишем начальное значение $x_0 \in [0; 1]$ в двоичной системе:

$x_0 = 0, a_1 a_2 a_3 \dots a_n \dots$, где $a_1 a_2 a_3 \dots a_n \dots$ - последовательность из нулей и единиц.

Или $x_0 = \frac{a_1}{2} + \frac{a_2}{2^2} + \frac{a_3}{2^3} + \dots$. Тогда, $2 * x_0 = a_1 + \frac{a_2}{2} + \frac{a_3}{2^2} + \dots$

То есть, последовательность нулей и единиц сдвигается влево на одну позицию.

Если $a_1 \neq 0$, то после операции (mod 1) - взятие дробной части, целая часть обратится в ноль, и мы получим: $x_1 = 0, a_2 a_3 \dots a_n \dots$

Такое преобразование двоичной последовательности, состоящее в сдвиге всех символов на одну позицию, называется сдвигом Бернулли. В нашем случае цифра 0 на первой позиции после разделительной запятой соответствует расположению числа в левой половине единичного интервала, 1 - в правой.

Если в качестве начального значения взять случайную последовательность, полученную, например, подбрасыванием монеты, то изображающая точка будет посещать левую и правую половину интервала в соответствии с полученной последовательностью нулей и единиц (0 - орел, 1 - решка). Изображающая точка следует нашей случайной последовательности. То есть это хаос.

Если в нулевом приближении x_0 достоверно известно только первые n знаков после запятой, то после n итераций все известные знаки исчерпываются, то есть, два начальных значения, отличных в $n + 1$ знаке порождают после n итераций последовательности, отличающиеся в первом знаке после запятой. Если двоичная дробь периодическая, что соответствует рациональному числу, то изображающая точка будет периодически повторять исходную точку x_0 через число временных шагов, равное периоду двоичного кода x_0 . Следовательно, такое отображение обладает бесконечным множеством периодических орбит (циклов).

Иррациональным числам x_0 соответствуют непериодические двоичные дроби. Следовательно, отображение имеет бесконечное число непериодических траекторий.

Подобное сложное поведение наблюдается и у непрерывного треугольного отображения (отображения «тент») рис.5.

$$x_{n+1} = \begin{cases} 2x_n & x \in [0; 1/2] \\ 2(1 - x_n) & x \in [1/2; 1] \end{cases}$$

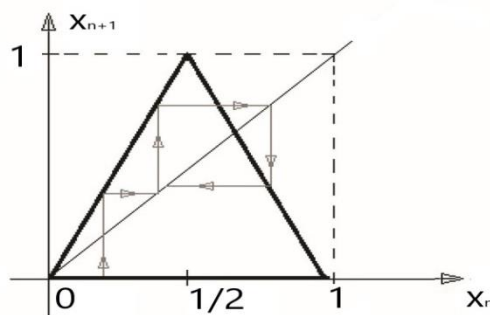


Рис. 5. Отображение «тент» - треугольное

Здесь также как и для отображения «зуб пилы», умножение на два, в случае если $x \in [0; 1/2]$ (старший разряд после запятой есть 0), приводит к сдвигу двоичного

кода на один разряд влево. Если же старший разряд равен 1, то после сдвига все оставшиеся биты инвертируются (0 заменяется на 1, 1 на 0). Аналогично отображению «зуб пилы» можно построить траекторию, посещающую левую и правую части графика в любой последовательности.

Логистическое отображение

Логистическое отображение, или его еще называют квадратичным отображением, часто использовалось в биологии в качестве простой модели биологической популяции. Динамика такого отображения была исследована Мэем [3], Фейгенбаумом [4], Лихтенбергом, Либерманом [5] и другими авторами.

Задается это отображение соотношением $x_{n+1} = rx_n(1 - x_n)$. Производная для этого отображения имеет вид $f'(x) = r - 2rx$.

где $r \geq 0$ - параметр, принадлежащий промежутку $[0;4]$, так что интервал $[0;1]$ отображается в себя. От параметра r зависит характер динамики нашего отображения.

При малых $r < 1$ у отображения только одна неподвижная точка $x = 0$. Чтобы проверить точку на устойчивость необходимо вычислить производную функции $f(x)$ при $x_0 = 0$: $f'(0) = r < 1$ и $x = 0$ устойчивая неподвижная точка. Т.о. в этом случае всегда $x_n \rightarrow 0$ при $n \rightarrow \infty$.

Компьютерные вычисления показывают, что в диапазоне $1 < r < 3$ x_n достигает некоторого ненулевого устойчивого состояния. Т.е. у отображения появилась устойчивая неподвижная точка рис. 6.

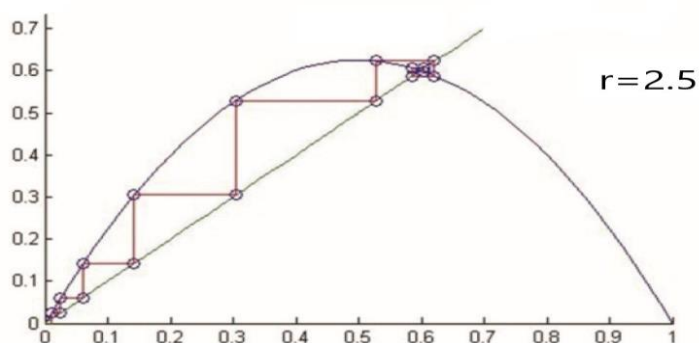


Рис. 6. Паутинная диаграмма логистического отображения при $r=2.5$

При дальнейшем увеличении параметра r , скажем до $r=3.3$, поведение x_n изменяется снова. Теперь, начиная с достаточно больших n , x_n колеблется около бывшего стационарного состояния, чередуясь между большим и меньшим значениями в соседних итерациях. Такой тип колебаний, в котором x_n повторяется каждые две итерации, мы называли циклом периода два.

При еще больших r (например, при $r = 3.5$) появляется цикл кратности 4 - x_n повторяется каждые четыре итерации. Т.о. предыдущий цикл удвоил свой период до периода 4.

Увеличивая r далее, получаем циклы периодов 8, 16, 32, ... Компьютерный эксперимент показывает, что справедливы следующие результаты, замеченные Митчелом Фейгенбаумом в 1976 году.

N	Тип цикла	Значение параметра r_N
1	2-цикл	3
2	4-цикл	3.44948
3	8-цикл	3.54408
4	16-цикл	3.56872
5	32-цикл	3.5698912
∞	Апериодический аттрактор	3.569945672

Заметим, что последовательные бифуркации происходят все быстрее и быстрее. В конечном счете, r_N сходится к предельному значению r_∞ . Фейгенбаум также нашел, что $\delta = \lim_{n \rightarrow \infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} = 4,669$.

Полученные численные результаты геометрически иллюстрируются на рис. 7.

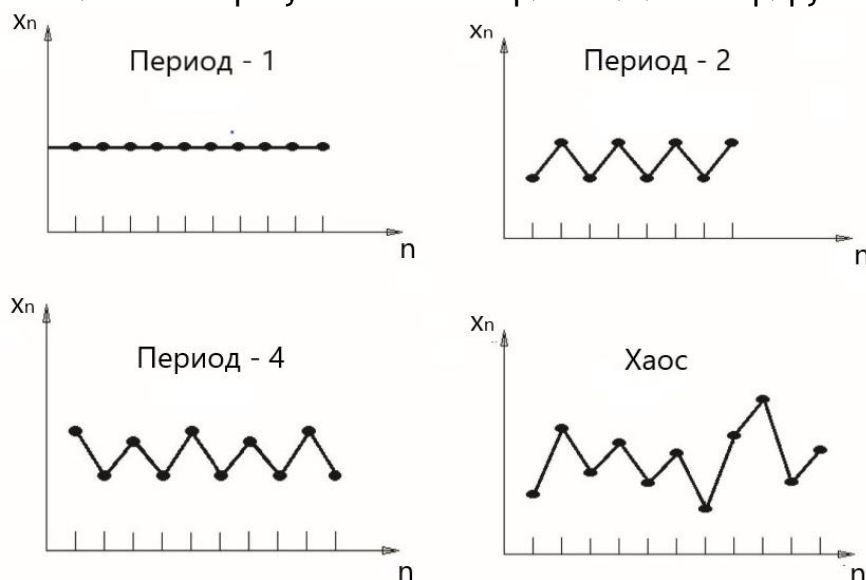


Рис. 7. Возможные типы решений логистического отображения

Позже, для отображения вида $x_{n+1} = \gamma \sin \pi x_n$ Митчел Фейгенбаум решил численно определить значения γ , при которых наблюдались бифуркации удвоения периода. Значения параметра также сходились геометрически. “... Что было совершенной неожиданностью, скорость сходимости выражалась тем же числом - 4.669” – Пишет ученый в своей работе “Универсальность в поведении нелинейных систем”. Данное значение (4.6692016091...) получило название - постоянная Фейгенбаума. Именно такой является скорость перехода к беспорядку всех динамических систем, испытывающих удвоение периода, например рост популяции, турбулентность.

Что же происходит при $r > r_\infty$. Чтобы увидеть поведение динамической системы сразу для всех значений параметра r , построим так называемую бифуркационную диаграмму, показанную на рис. 8. На ней после переходного процесса откладывают значения x_n над текущим значением r . Переходя к соседнему значению r , повторяют процедуру до $r = 4$.

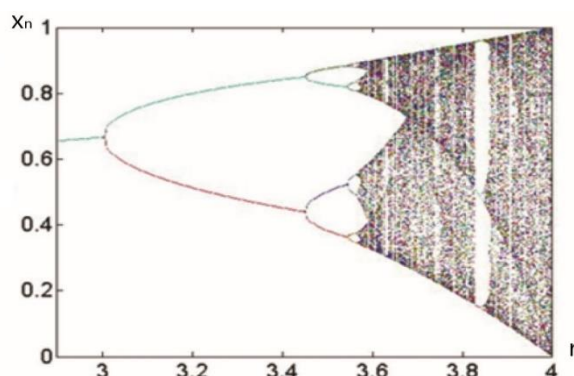


Рис. 8. Бифуркационная диаграмма логистического отображения

На горизонтальной оси отложены значения параметра r в промежутке $[2.9; 4]$, по вертикальной оси периодические точки, соответствующие данному значению параметра. Значения параметра берется, начиная с числа 2.9 для того, чтобы исключить

предварительные итерации, при которых не происходит динамики удвоения периода. Разделение линии на две части, потом еще на 2 и так далее символизирует удвоение периода. Для наглядности линии выделены разными цветами. При увеличении численного значения параметра число периодических точек стремится к бесконечности. Видно, что после 8 кратного цикла зрительно проследить удвоение не получается, линии смешиваются в единое целое. Но потом появляются вкрапления своеобразных “окон периодичности”. Наблюдается смесь порядка и хаоса. Большое окно начинается около $r = 3,83$ и содержит устойчивые циклы периода 3.

Одним из открытий Фейгенбаума стало то, что бифуркационное дерево в окрестности перехода к хаосу обладает свойством скейлинга, то есть самоподобной структурой. Для иллюстрации данного свойства на рис. 9 изображены два фрагмента бифуркационного дерева. На одном из них логистическое отображение рассматривается в промежутке параметра от 3.4 до 4, на другом выделена область в промежутке от 3.848 до 3.856 сразу же за окном периодичности периода три.

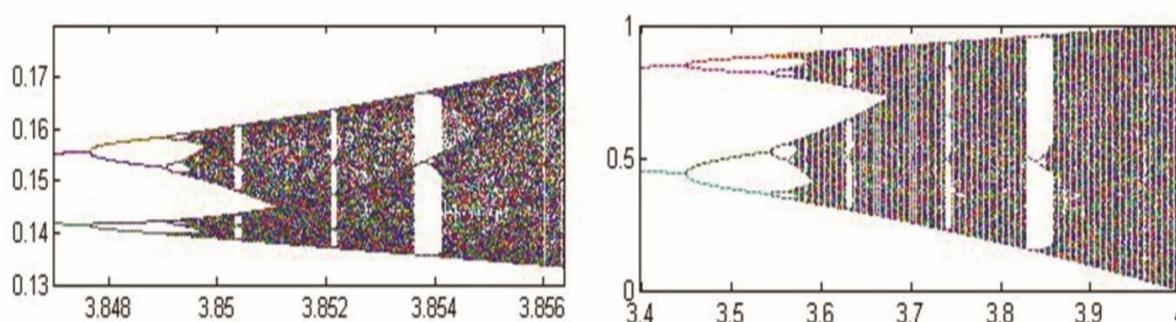


Рис. 9. Самоподобие логистического отображения

Практически полная копия в миниатюре.

Заключение

В работе исследовано поведение одномерных непрерывных отображений отрезка в себя. Численным моделированием получен каскад бифуркаций удвоения периода, построена бифуркационная диаграмма для логистического отображения.

ССЫЛКИ НА ИСТОЧНИКИ

1. Андронов А.А., Витт А.А., Хайкин С.Э. Теория колебаний. 2-е изд. М.: Физматгиз, 1959. 926с.
2. Steven Strogatz, «Non-linear Dynamics and Chaos: With applications to Physics, Biology, Chemistry and Engineering», Perseus Books, 2000.
3. May R.M. Simple Mathematical Models with very Complicated Dynamics. Nature 261, 459 (1976)
4. Фейгенбаум М. Универсальность и поведение нелинейных систем. – Успехи физ. Наук, 1983, с. 137-145.
5. Лихтенберг А., Либерман М. Регулярная и стохастическая динамика. М.: Мир, 1984. 536с.

Gribov A.F.

Cand. Sci (Phis.-Math.), Assoc. Professor of Bauman Moscow State Technical University
alexandr-gribov@list.ru

Krasnov I.K.

Cand. Sci (Eng.) Assoc. Professor of Bauman Moscow State Technical University
igorkrsnv@yandex.ru

Elementary introduction to the theory of one-dimensional discrete dynamical system

Abstract. The article presents an elementary introduction to the theory of dynamical systems on the example of their simplest representatives-one-dimensional discrete maps. The basic concepts arising in the study of such systems are considered: fixed points, their stability, orbit, determinism, chaos, etc. it is intended for high school students and first-year students for the initial independent acquaintance with a new rapidly developing mathematical discipline - mathematical dynamics.

Keywords: one-dimensional mapping, trajectory, fixed point, chaos.

МЕТОДИЧЕСКИЕ АСПЕКТЫ ИЗЛОЖЕНИЯ ТЕМЫ «ИНТЕГРИРОВАНИЕ ТРИГОНОМЕТРИЧЕСКИХ ФУНКЦИЙ» В КУРСЕ «НЕОПРЕДЕЛЕННЫЙ ИНТЕГРАЛ»

Аннотация

В статье рассмотрены основные правила интегрирования функций, рационально зависящих от тригонометрических функций $\sin x, \cos x$. Все они проиллюстрированы примерами. В работе приведены задачи, при решении которых использованы различные приемы, что позволяет провести сравнение этих методов и вспомнить некоторые теоретические моменты. В качестве итога предложен алгоритм интегрирования тригонометрических функций.

Ключевые слова

первообразная, интеграл, универсальная подстановка, простейшие дроб

АВТОРЫ

Ефремова Светлана Николаевна,

старший преподаватель

ФГБОУ ВО «Московский государственный технический университет

им. Н. Э. Баумана», г. Москва

Efremova.sn@gmail.com

Косова Анна Владимировна,

старший преподаватель

ФГБОУ ВО «Московский государственный технический университет

им. Н. Э. Баумана», г. Москва

Anna.v.kosova@mail.ru

Ласковая Татьяна Алексеевна,

старший преподаватель

ФГБОУ ВО «Московский государственный технический университет

им. Н. Э. Баумана», г. Москва

talaskovy@mail.ru

Введение

Интегрирование, т.е. восстановление функции по ее производной, - важнейшая операция, без которой невозможно изучение высшей математики. Научиться интегрировать непросто. И хотя не вызывает сомнения тот факт, что овладеть интегрированием можно, только решив большое количество примеров, важность роли преподавателя в процессе обучения он не уменьшает. Ведь в отличие от обратной задачи, дифференцирования, в интегрировании нет полного набора четко сформулированных правил, следование которым обеспечивает решение любого примера. Это усложняет работу преподавателя, который должен не только познакомить учащихся с основными рекомендациями, но и помочь выбрать правильное направление для их действий.

В предложенной работе демонстрируются различные подходы к решению примеров одного из наиболее сложных разделов интегрирования - интегралов от тригонометрических функций. Подобные примеры позволяют сравнивать методы, отмечая слабые и сильные стороны, и анализировать полученные результаты.

Статья будет полезна не только преподавателям, которые смогут использовать ее при подготовке и проведении занятий, но и студентам, расширяя их кругозор и возможности в изучении сложной темы.

Методология и результаты исследований

Одним из главных инструментов при интегрировании рациональных функций от $\sin x, \cos x: R(\sin x, \cos x)$, является универсальная тригонометрическая подстановка:

$t = \operatorname{tg} \frac{x}{2}$ [1]. Ее универсальность заключается в том, что она заменяет исходный интеграл от тригонометрической функции на равный ему интеграл от рациональной дроби, для которого известен алгоритм решения - разложение подынтегральной функции на простейшие дроби и применение свойств неопределенного интеграла.

$$\text{Имеем: } \sin x = \frac{2t}{1+t^2}, \cos x = \frac{1-t^2}{1+t^2}, dt = \frac{2dt}{1+t^2}.$$

Применение этой подстановки часто приводит к громоздким вычислениям, но она неизменно является успешной для интегралов вида

$$\int \frac{dx}{a \sin x + b \cos x + c}; \int \frac{dx}{\cos^{2k+1} x}; \int \frac{dx}{\sin^{2k+1} x}, k \in \mathbb{Z}.$$

Рассмотрим несколько примеров ее применения.

Пример 1. Вычислить $\int \frac{dx}{1 + \cos x - 5 \sin x}$.

Решение:

$$\begin{aligned} \int \frac{dx}{1 + \cos x - 5 \sin x} &= \left. \begin{array}{l} t = \operatorname{tg} \frac{x}{2} \\ \sin x = \frac{2t}{1+t^2} \\ \cos x = \frac{1-t^2}{1+t^2} \\ dx = \frac{2dt}{1+t^2} \end{array} \right| = \int \frac{2dt}{(1+t^2) + (1-t^2) - 10t} = \int \frac{dt}{1-5t} = -\frac{1}{5} \int \frac{d(1-5t)}{1-5t} = \\ &= -\frac{1}{5} \ln |1-5t| + C = -\frac{1}{5} \ln \left| 1 - 5 \operatorname{tg} \frac{x}{2} \right| + C \end{aligned}$$

Пример 2. Вычислить $\int \frac{dx}{\sin^3 x}$.

$$\begin{aligned} \int \frac{dx}{\sin^3 x} &= \int \frac{\frac{2dt}{1+t^2}}{\left(\frac{2t}{1+t^2}\right)^3} = \frac{1}{4} \int \frac{(1+t^2)^2 dt}{t^3} = \frac{1}{4} \int \frac{1+2t^2+t^4}{t^3} dt = \frac{1}{4} \int \left(\frac{1}{t^3} + \frac{2}{t} + t \right) dt = \\ &= \frac{1}{4} \left(-\frac{1}{2t^2} + 2 \ln |t| + \frac{t^2}{2} \right) + C = -\frac{1}{8 \operatorname{tg}^2 \frac{x}{2}} + \frac{1}{2} \ln \left| \operatorname{tg} \frac{x}{2} \right| + \frac{1}{8} \operatorname{tg}^2 \frac{x}{2} + C \end{aligned}$$

Очень полезно для сравнения применить универсальную подстановку и для решения следующего примера.

Пример 3. Вычислить $\int \frac{dx}{\cos^3 x}$.

$$\text{Решение: } \int \frac{dx}{\cos^3 x} = \int \frac{\frac{2dt}{1+t^2}}{\left(\frac{1-t^2}{1+t^2}\right)^3} = 2 \int \frac{(1+t^2)^2 dt}{(1-t^2)^3} = 2 \int \frac{(1+t^2)^2 dt}{(1-t)^3 (1+t)^3}$$

Далее потребуется разложить подинтегральную правильную рациональную дробь на простейшие дроби, которых будет шесть. Т.о. универсальная тригонометрическая подстановка в этом примере приводит к более трудоемкому решению, т.к. для вычисления неопределенных коэффициентов потребуется решить СЛАУ из шести уравнений с шестью неизвестными.

Естественным образом возникает стремление найти в качестве альтернативы более легкие пути интегрирования. Основные практические итоги поисков можно сформулировать в виде следующих правил [2].

Правило 1. Если подинтегральная функция нечетная относительно $\sin x$: $R(-\sin x, \cos x) = -R(\sin x, \cos x)$, то применяют подстановку $t = \cos x$.

Правило 2. Если подинтегральная функция нечетная относительно $\cos x$: $R(\sin x, -\cos x) = -R(\sin x, \cos x)$, то применяют подстановку $t = \sin x$.

Правило 3. Если функция четная по совокупности: $R(-\sin x, -\cos x) = R(\sin x, \cos x)$, то применяют подстановку $t = \operatorname{tg} x$ или $t = \operatorname{ctg} x$. Выбирают ту, которая удобна в каждом конкретном случае.

Правило 4. Если функция четная по совокупности функций: $R(-\sin x, -\cos x) = R(\sin x, \cos x)$ и степени $\sin x, \cos x$, как правило, неотрицательные, то применяют формулы понижения степени:
 $\cos^2 x = \frac{1 + \cos 2x}{2}$, $\sin^2 x = \frac{1 - \cos 2x}{2}$, $\sin x \cos x = \frac{1}{2} \sin 2x$.

Рассмотренный ранее пример 3 можно решить, используя правило 2:

$$\begin{aligned} \int \frac{dx}{\cos^3 x} &= \int \frac{\cos x dx}{\cos^4 x} = \int \frac{d(\sin x)}{(1 - \sin^2 x)^2} = \int \frac{dt}{(1-t)^2(1+t)^2} = \frac{1}{4} \int \left(\frac{1}{1-t} + \frac{1}{(1-t)^2} + \frac{1}{1+t} + \frac{1}{(1+t)^2} \right) dt = \\ &= \frac{1}{4} \left(-\ln|1-t| + \ln|1+t| + \frac{1}{1-t} + \frac{1}{1+t} \right) + C = \frac{1}{4} \left(\ln \left| \frac{1-\sin x}{1+\sin x} \right| + \frac{1}{1-\sin x} - \frac{1}{1+\sin x} \right) + C \end{aligned}$$

Вычисляя интеграл, используя правило 2, мы смогли несколько упростить решение данной задачи.

Кроме того, интегрируя тригонометрические функции, не следует забывать основное тригонометрическое тождество. Так, рассмотренный выше пример 3, для которого универсальная тригонометрическая подстановка приводит к громоздким вычислениям, довольно просто решается с помощью основного тригонометрического тождества и интегрирования по частям:

$$\int \frac{dx}{\cos^3 x} = \int \frac{\sin^2 x + \cos^2 x}{\cos^3 x} dx = \int \frac{\sin^2 x}{\cos^3 x} dx + \int \frac{dx}{\cos x} = \left. \begin{array}{l} u = \sin x \\ du = \cos x dx \\ dv = \frac{\sin x dx}{\cos^3 x} \\ v = \frac{-d(\cos x)}{\cos^3 x} = \frac{1}{2 \cos^2 x} \end{array} \right| =$$

$$= \frac{\sin x}{2 \cos^2 x} - \frac{1}{2} \int \frac{dx}{\cos x} + \int \frac{dx}{\cos x} = \frac{\sin x}{2 \cos^2 x} + \frac{1}{2} \ln \left| \operatorname{tg} \left(\frac{x}{2} + \frac{\pi}{4} \right) \right| + C$$

Сравним применение разных способов еще на одном примере.

Пример 4. Найти $\int \frac{dx}{\sin x \cos x}$.

Подинтегральная функция нечетная относительно $\sin x$, нечетная относительно $\cos x$, четная по совокупности этих функций.

1) Применим универсальную подстановку:

$$\int \frac{dx}{\sin x \cos x} = \int \frac{2dt}{\frac{1+t^2}{2t} \frac{1-t^2}{1+t^2}} = \int \frac{1+t^2}{t(1-t^2)} dt = \int \frac{1+t^2}{t(1-t)(1+t)} dt = \int \left(\frac{1}{t} + \frac{1}{1-t} - \frac{1}{1+t} \right) dt =$$

$$= \ln|t| - \ln|1-t| - \ln|1+t| + C = \ln \left| \frac{t}{1-t^2} \right| + C = \ln \left| \frac{\operatorname{tg} \frac{x}{2}}{1 - \operatorname{tg}^2 \frac{x}{2}} \right| + C$$

2) Согласно правила 1 замена $t = \cos x$:

$$\int \frac{dx}{\sin x \cos x} = \int \frac{\sin x dx}{\sin^2 x \cos x} = \left| \begin{array}{l} t = \cos x \\ dt = -\sin x dx \end{array} \right| = - \int \frac{dt}{(1-t^2)t} = \int \left(\frac{1/2}{t-1} + \frac{1/2}{t+1} - \frac{1}{t} \right) dt =$$

$$= \frac{1}{2} \ln|t-1| + \frac{1}{2} \ln|t+1| - \ln|t| + C = \frac{1}{2} \ln \left| \frac{t^2-1}{t^2} \right| + C = \frac{1}{2} \ln \left| \frac{\cos^2 x - 1}{\cos^2 x} \right| + C = \frac{1}{2} \ln \left| \frac{\sin^2 x}{\cos^2 x} \right| + C = \frac{1}{2} \ln \operatorname{tg}^2 x + C$$

3) Согласно правила 2 замена $t = \sin x$:

$$\int \frac{dx}{\sin x \cos x} = \int \frac{\cos x dx}{\sin x \cdot \cos^2 x} = \left| \begin{array}{l} t = \sin x \\ dt = \cos x dx \end{array} \right| = \int \frac{dt}{t(1-t^2)} = \int \left(\frac{1}{t} + \frac{1/2}{1-t} - \frac{1/2}{1+t} \right) dt =$$

$$= \ln|t| - \frac{1}{2} \ln|1-t| - \frac{1}{2} \ln|1+t| + C = \frac{1}{2} \ln \left| \frac{t^2}{1-t^2} \right| + C = \frac{1}{2} \ln \left| \frac{\sin^2 x}{1-\sin^2 x} \right| + C = \frac{1}{2} \ln \operatorname{tg}^2 x + C$$

4) Согласно правила 3 замена $t = \operatorname{tg} x$:

$$\int \frac{dx}{\sin x \cos x} = \int \frac{\cos x}{\sin x} \frac{dx}{\cos^2 x} = \int \frac{d(\operatorname{tg} x)}{\operatorname{tg} x} = \ln|\operatorname{tg} x| + C = \frac{1}{2} \ln \operatorname{tg}^2 x + C$$

или $t = \operatorname{ctg} x$: $\int \frac{dx}{\sin x \cos x} = \int \frac{\sin x}{\sin^2 x} \frac{dx}{\cos x} = - \int \frac{d(\operatorname{ctg} x)}{\operatorname{ctg} x} = -\ln|\operatorname{ctg} x| + C = \frac{1}{2} \ln \operatorname{tg}^2 x + C$

5) В данном примере можно понизить степень в знаменателе, используя удвоенный аргумент (правило 4).

Тогда получим:

$$\int \frac{dx}{\sin x \cos x} = \left| \sin x \cos x = \frac{1}{2} \sin 2x \right| = \int \frac{2dx}{\sin 2x} = \int \frac{d(2x)}{\sin 2x} = \ln |tg x| + C = \frac{1}{2} \ln tg^2 x + C.$$

В зависимости от выбора пути решения получаем различные формы записи ответа - через разные тригонометрические функции. По свойству первообразной известно, что две первообразные одной функции отличаются друг от друга на константу. Убедимся в этом, преобразовав ответ 1 этого примера:

$$\ln \left| \frac{tg \frac{x}{2}}{1 - tg^2 \frac{x}{2}} \right| = \ln \left| \frac{1}{2} \frac{2tg \frac{x}{2}}{1 - tg^2 \frac{x}{2}} \right| = \ln \frac{1}{2} + \ln |tg x| = \ln \frac{1}{2} + \frac{1}{2} \ln tg^2 x.$$

Следует обратить внимание студентов на возможность разных форм записи одного и того же ответа. Если полученный обучающимся результат не совпадает с ответом в задачнике, это не обязательно говорит об ошибке. Возможно, это означает, что ответ в задачнике получен другим методом. Это относится не только к нахождению первообразных тригонометрических функций, но и к интегрированию в целом.

При решении примера 4 (пункт 5) мы воспользовались формулой синуса двойного аргумента. Однако, следует отметить, что формулы понижения степени крайне редко применяются в знаменателе. Как правило, они применяются при вычислении интегралов вида $\int \sin^m x \cos^n x dx$, где $m = 0, 1, \dots; n = 0, 1, \dots; m + n = 2k, k \in \mathbb{Z}$. Рассмотрим пример.

Пример 5. Найти $\int \sin^2 x \cos^4 x dx$.

Применим различные способы решения этой задачи:

$$\begin{aligned} 1) \int \sin^2 x \cos^4 x dx &= \int \frac{1 - \cos 2x}{2} \left(\frac{1 + \cos 2x}{2} \right)^2 dx = \frac{1}{8} \int (1 - \cos 2x)(1 + 2\cos 2x + \cos^2 2x) dx = \\ &= \frac{1}{8} \int (1 + \cos 2x - \cos^2 2x - \cos^3 2x) dx = \frac{1}{8} \left(x + \frac{1}{2} \sin 2x - \int \frac{1 + \cos 4x}{2} dx - \int (1 - \sin^2 2x) \cos 2x dx \right) = \\ &= \frac{1}{8} x + \frac{1}{16} \sin 2x - \frac{1}{16} x - \frac{1}{64} \sin 4x - \frac{1}{16} \sin 2x + \frac{1}{48} \sin^3 2x + C = \frac{1}{16} x - \frac{1}{64} \sin 4x + \frac{1}{48} \sin^3 2x + C \\ 2) \int \sin^2 x \cos^4 x dx &= \frac{1}{4} \int (2 \sin x \cos x)^2 \cos^2 x dx = \frac{1}{8} \int \sin^2 2x (1 + \cos 2x) dx = \\ &= \frac{1}{8} \int \left(\frac{1 - \cos 4x}{2} + \sin^2 2x \cos 2x \right) dx = \frac{1}{16} x - \frac{1}{64} \sin 4x + \frac{1}{48} \sin^3 2x + C. \end{aligned}$$

Как видно, второй вариант решения оказался более коротким.

И, в заключение, еще одно правило.

Правило 5. Если под знаком интеграла - произведение тригонометрических функций разных аргументов, то необходимо преобразовать произведение в сумму, используя формулы:

$$\sin nx \cos mx = \frac{1}{2} (\sin(n-m)x + \sin(n+m)x)$$

$$\sin nx \sin mx = \frac{1}{2} (\cos(n-m)x - \cos(n+m)x)$$

$$\cos nx \cos mx = \frac{1}{2} (\cos(n-m)x + \cos(n+m)x)$$

Пример 6. Найти $\int \sin 3x \cos 5x dx$.

Выбираем нужную формулу и получаем:

$$\int \sin 3x \cos 5x dx = \frac{1}{2} \left(\int \sin(-2x) + \sin 8x \right) = \frac{1}{4} \cos 2x - \frac{1}{16} \cos 8x + C.$$

Интегрирование тригонометрических функций - одна из наиболее сложных тем курса. Попробуем сформулировать алгоритм, который сможет помочь при решении задач этого раздела.

1. Проверить, применимо ли для данного интеграла правило 5. Если нет, то перейти к следующему пункту.

2. Проверить, является ли данный интеграл интегралом вида $\int \sin^m x \cos^n x dx$, где $m = 0, 1, \dots; n = 0, 1, \dots; m + n = 2k, k \in \mathbb{N}$. Если да - применить формулы понижения степени. Если нет - перейти к пункту 3.

3. Проверить, применимы ли к вычислению интеграла правила 1-4. Если да - выбрать подходящее и следовать ему.

4. Если ни одно из правил не подходит - применить универсальную тригонометрическую подстановку.

Заключение

Интегрирование тригонометрических функций - тема очень многообразная. Поэтому, в рамках одной статьи, невозможно разобрать все связанные с ней трудности, и ответить на все вопросы, которые могут возникнуть при вычислении интегралов. Предметом рассмотрения были основные правила, которые используются при нахождении первообразных тригонометрических функций. В статье были продемонстрированы различные подходы к одним и тем же примерам, что позволяет сравнить их и выбрать более рациональный способ решения задачи. В итоге, был предложен алгоритм интегрирования функций, рационально зависящих от $\sin x, \cos x$.

ССЫЛКИ НА ИСТОЧНИКИ

1. Кудрявцев Л.Д. Курс математического анализа: в 3 т., т. 1 – М.: Юрайт, 2014.
 2. Зарубин В.С., Иванова Е.Е., Кувыркин Г.Н. Интегральное исчисление функций одного переменного: Учебник для студентов вузов/ Под ред. В.С.Зарубина, А.П.Крищенко. – М.: Изд-во МГТУ им. Н.Э.Баумана, 1999.
-

Svetlana Efremova

Senior lecturer, Bauman Moscow State Technical University, Moscow

Efremova.sn@gmail.com

Anna Kosova

Senior lecturer, Bauman Moscow State Technical University, Moscow

anna.v.kosova@mail.ru

Tatyana Laskovaya

Senior lecturer, Bauman Moscow State Technical University, Moscow

talaskovy@mail.ru

Methodological aspects of the presentation of the topic "Integration of trigonometric functions" in the course "Indefinite integral"

Abstract. The article considers the basic rules for the integration of functions rationally depended on trigonometric functions. They are all illustrated by examples. The paper presents problems for the solution of which various techniques were used, that allows to compare these methods and recall some theoretical points. As a result, an algorithm for integrating trigonometric functions is proposed.

Keywords: antiderivative, integral, universal substitution, simple fractions.

ОБ ИНФОРМАТИЗАЦИИ ПРОВЕРКИ ДОМАШНЕГО ЗАДАНИЯ ПО КРИВЫМ И ПОВЕРХНОСТЯМ ВТОРОГО ПОРЯДКА

Аннотация

Важность информатизации образования в современном обществе трудно переоценить. Информационные технологии позволяют вывести процесс обучения на новый уровень. В исследованиях, посвящённых этому вопросу, много внимания уделяется процессу передачи информации обучаемому и значительно меньше - обратной связи, позволяющей контролировать процесс усвоения материала обучаемым. Данная статья восполняет этот пробел, рассказывая об опыте автора по информатизации этой области учебного процесса. Целью исследования было изучение возможности информатизации такого рода и эффективности применяемых средств. Автором статьи было разработано программное обеспечение, эффективность которого была проверена на практике.

Ключевые слова

информатизация образования, программное обеспечение, линейная алгебра

АВТОР

Кавинов Алексей Владимирович,
доцент кафедры математического моделирования
ФГБОУ ВО «Московский государственный технический университет
им. Н. Э. Баумана», г. Москва
alekseyvladimirovich1@yandex.ru

Введение

Время, в которое мы живём, характеризуется неудержимым развитием технического прогресса. Компьютеризация, роботизация, использование нанотехнологий и искусственного интеллекта стали неотъемлемыми чертами практически всех областей современной жизни.

На необходимость информатизации учебного процесса не раз указывалось руководством Российской Федерации, в частности, министр образования Ольга Васильева заявила, что "российские школы и вузы находятся на пороге цифровой революции, и преподаватели должны быть к ней готовы. Революция неизбежна ... отторжения учителя от новых технологий быть не должно" [1]. Неизбежность информатизации, её задачи и проблемы стали темой массы статей различных авторов [2-10], что лишний раз показывает актуальность вопроса.

Несмотря на всю важность информатизации, далеко не везде в учебном процессе в полной мере используются новые возможности современных технологий. Говоря об информатизации, обычно имеют в виду процесс передачи знаний студенту при помощи современных технологий и значительно реже - оптимизацию передачи и анализа данных в обратном направлении; тем не менее, наличие обратной связи при обучении очень важно. Вовремя обнаруженное недопонимание студентом различных аспектов изучаемой дисциплины позволяет скорректировать когнитивные процессы, предотвратив растущее непонимание в дальнейшем. Тесты позволяют существенно

оптимизировать процесс оценки знаний, выявить пробелы в знаниях, однако как средство контроля освоения материала они имеют тот недостаток, что позволяют контролировать только конечный результат решения задачи, игнорируя причины неудачи. Не поддаётся контролю при помощи тестов, в частности, ход логических рассуждений, приведший к тому или иному ответу [11,12]. В связи с этим представляется, что «классические» домашние задания рано списывать со счетов там, где важным является не получение ответа как самоцель, но развитие образа мышления, присущего конкретной дисциплине.

Методология и результаты исследования

Домашние задания являются важной частью учебного процесса в МГТУ им. Н.Э.Баумана. Выполняя домашние задания по математике, студенты могут вдумчиво, без суеты и волнений, присущих аудиторным контрольным работам, вникнуть в суть решаемых задач.

Проверка домашних заданий требует от преподавателя сосредоточенности и подчас немалых временных затрат. Оптимизация этого процесса при помощи ЭВМ позволит повысить долю ошибок, найденных преподавателем, исключить при проверке ошибки самого преподавателя и сократить время, затрачиваемое на процесс проверки. В этой статье автор хотел бы поделиться с читателями своим опытом информатизации этой части учебного процесса, полученным при работе в качестве преподавателя на кафедре ФН-12 «Математическое моделирование» МГТУ им. Н.Э.Баумана.

Домашнее задание «Кривые и поверхности второго порядка» является частью учебного плана по курсу «Линейная алгебра и функции нескольких переменных», входящего в программу первого курса обучения. Это задание содержит две задачи по кривым и две - по поверхностям второго порядка. В процессе его выполнения студенты закрепляют полученные знания по темам «квадратичные формы», «базисы», «матрицы перехода», «собственные числа и собственные векторы»; при этом также подразумевается проведение студентом некоторого объёма вычислений, в процессе которых тот может совершить ошибку. Ошибочный ответ может быть получен как в результате таких «арифметических» ошибок, так и в результате недопонимания лежащих в основе этих вычислений понятий и алгоритмов. Выяснение причин ошибок позволяет более точно определить, если таковые имеются, пробелы в знаниях конкретного студента. Это, однако, означает, что преподаватель должен частично или полностью повторять вычисления студента. Иногда это приводит к существенным затратам времени на проверку, ведь общее количество проверяемых заданий одного типа, приходящихся на преподавателя, может достигать 200 заданий за семестр.

Значительно упрощён процесс проверки может быть путём использования имеющихся в рукописном и компьютерном вариантах на кафедре ФН-12 ответов к домашним заданиям. Эти ответы, однако, обладают важным недостатком.

При выполнении домашнего задания ход действий студента может быть разным. Хотя все способы решения приводят в результате к одному и тому же правильному ответу, промежуточные выкладки при этом различаются. Так, в домашнем задании «Кривые и поверхности второго порядка» по курсу «Линейная алгебра и функции нескольких переменных» возможно 8 правильных вариантов решения в задачах по кривым второго порядка, а в задачах по поверхностям их число увеличивается до 48. Конечно, такое количество решений одной задачи не может быть отражено в имеющихся «статических» ответах. Решение этой проблемы лежит в плоскости создания нового интерактивного типа «динамических» ответов. Такие ответы представляют со-

бой компьютерные программы, позволяющие пользователю выбрать один из возможных вариантов решения и затем производящие вычисления и графические построения в соответствии с выбранным вариантом.

Автором этих строк в целях личного использования был разработан комплекс прикладных программ, включающий в себя три выполняемых файла: один для вычисления коэффициентов уравнений и отображения кривых второго порядка, второй - для вычисления коэффициентов преобразованных уравнений поверхностей второго порядка и третий - для вывода изображения поверхности на экран.

Программа «Кривые второго порядка» разработана при помощи бесплатной среды разработки Visual Studio Community. Языком разработки был выбран C#. Причиной такого выбора было наличие развитых средств разработки графического интерфейса. Графический пользовательский интерфейс был реализован на базе возможностей, предоставляемых интерфейсом Windows Forms.

Внешний вид окна программы показан на рисунке 1.

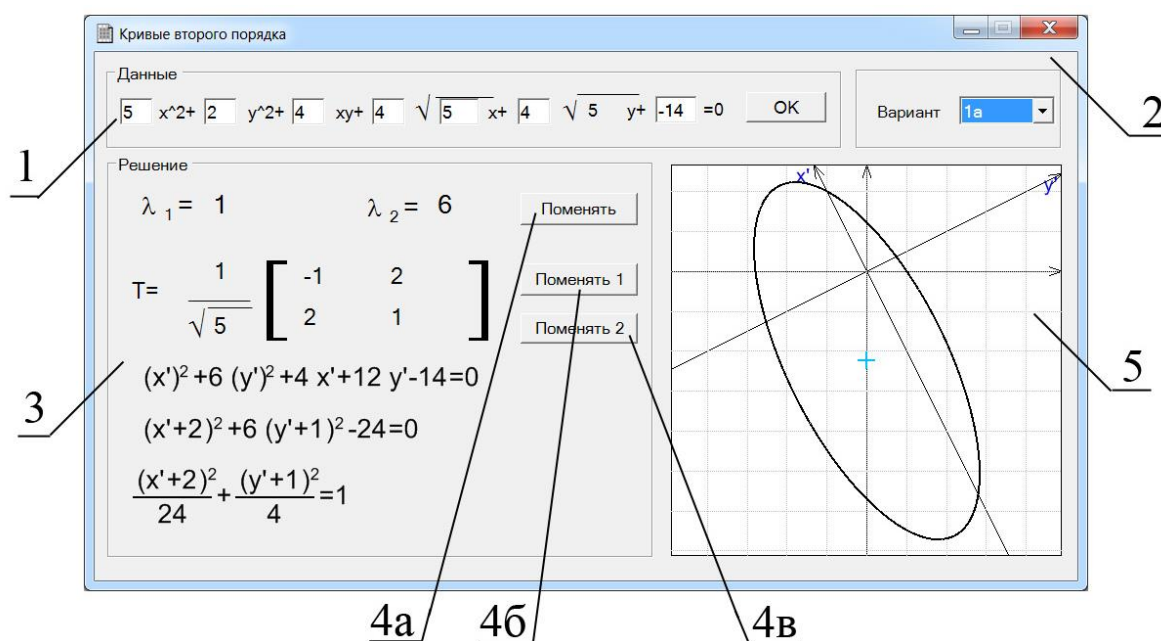


Рис. 1. Внешний вид окна программы «Кривые второго порядка»

Среди элементов интерфейса программы можно выделить область ввода данных 1, область выбора варианта 2, область решения 3 и область графических построений 5. В начале работы с программой пользователь может как задать вручную исходные данные для работы программы с помощью полей ввода области 1, так и, используя область 2, выбрать один из «вшитых» в программу вариантов заданий. После выбора набора исходных данных программа выводит текстовую часть решения в область 3, а графические построения - в область 5. Область 3 содержит информацию о собственных числах квадратичной формы и об ортогональном преобразовании, приводящем уравнение кривой к каноническому виду. В этой области отображается также вид уравнения кривой в новой системе координат до и после выделения полных квадратов и канонический вид уравнения кривой. Получение этих уравнений разделяет основные этапы выполнения первых двух задач домашнего задания, что позволяет при проверке быстро локализовать ошибку в студенческой работе. Область 5 содержит изображение заданной пользователем кривой второго порядка в исходной системе координат, изображения осей исходной и преобразованной систем координат и маркер центра кривой. Внешний вид этой области может быть настроен пользователем по

своему вкусу и желанию при помощи контекстного меню (см. рисунок 2). Масштаб рисунка и диапазон отображения по осям x и y выбираются автоматически таким образом, чтобы показать часть кривой, нужную для проверки.

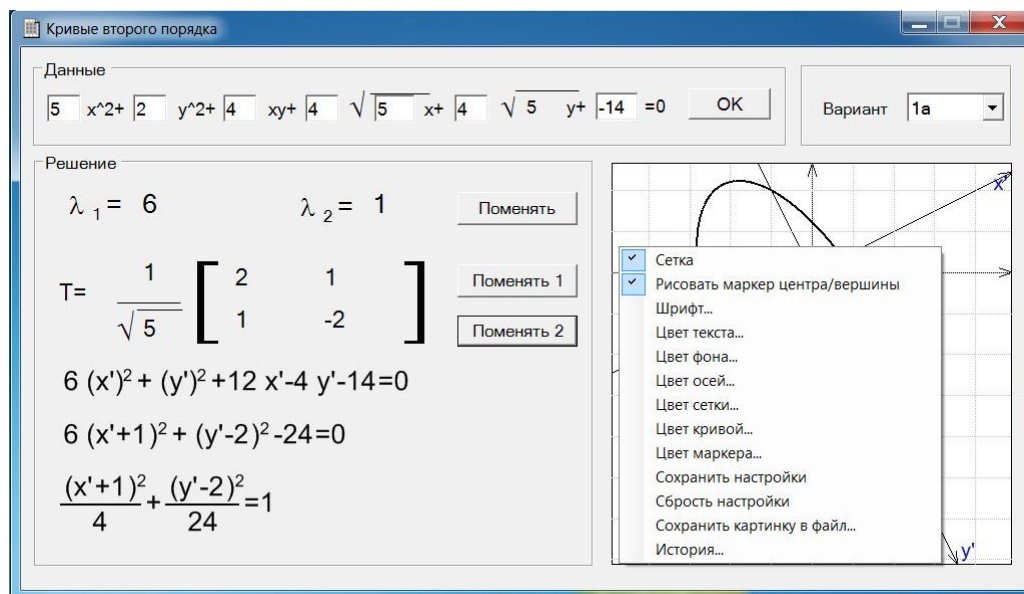


Рис. 2. Возможности настройки области графического вывода

Интерактивность процесса поиска решения обеспечивается элементами пользовательского интерфейса 4а, 4б и 4в. С помощью элемента 4а пользователь может менять порядок собственных чисел в квадратичной форме канонического вида, с помощью элементов 4б и 4в - знак первого и второго соответственно собственных векторов, отвечающих этим собственным числам. Таким образом можно получить результат, соответствующий любому из восьми возможных способов решения.

На рисунке 3 показан вид окна программы в момент отображения решения того же варианта с другим расположением собственных чисел и собственных векторов в матрице квадратичной формы канонического вида и, соответственно, в матрице ортогонального преобразования. Из рисунка видно, что, хотя ход решения отличается от показанного на рисунке 1, конечный результат вычислений (изображение эллипса в исходной системе координат) остался тем же.

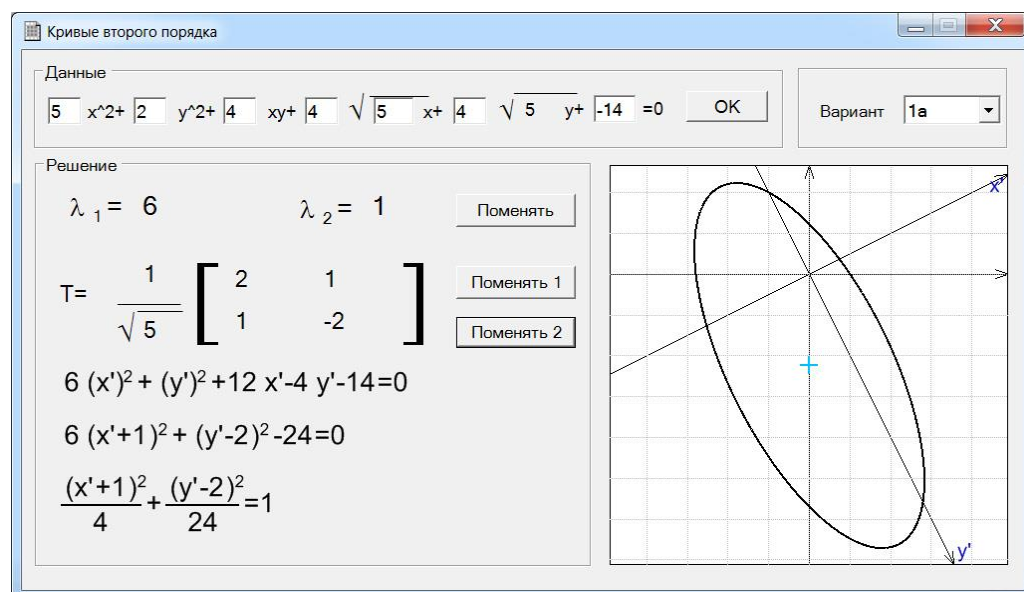


Рис. 3. Возможный внешний вид окна программы «Кривые второго порядка»

Вторая часть домашнего задания посвящена применению квадратичных форм к исследованию поверхностей второго порядка. Помочь в процессе её проверки могут две остальные программы разработанного программного комплекса.

Программа «Поверхности второго порядка» является первой из двух, предназначенных для проверки второй части домашнего задания. Она разработана, как и программа «Кривые второго порядка», при помощи бесплатной среды разработки Visual Studio Community. Языком разработки был выбран C#. Причиной такого выбора было наличие развитых средств разработки графического интерфейса. Внешний вид окна программы показан на рисунке 4.

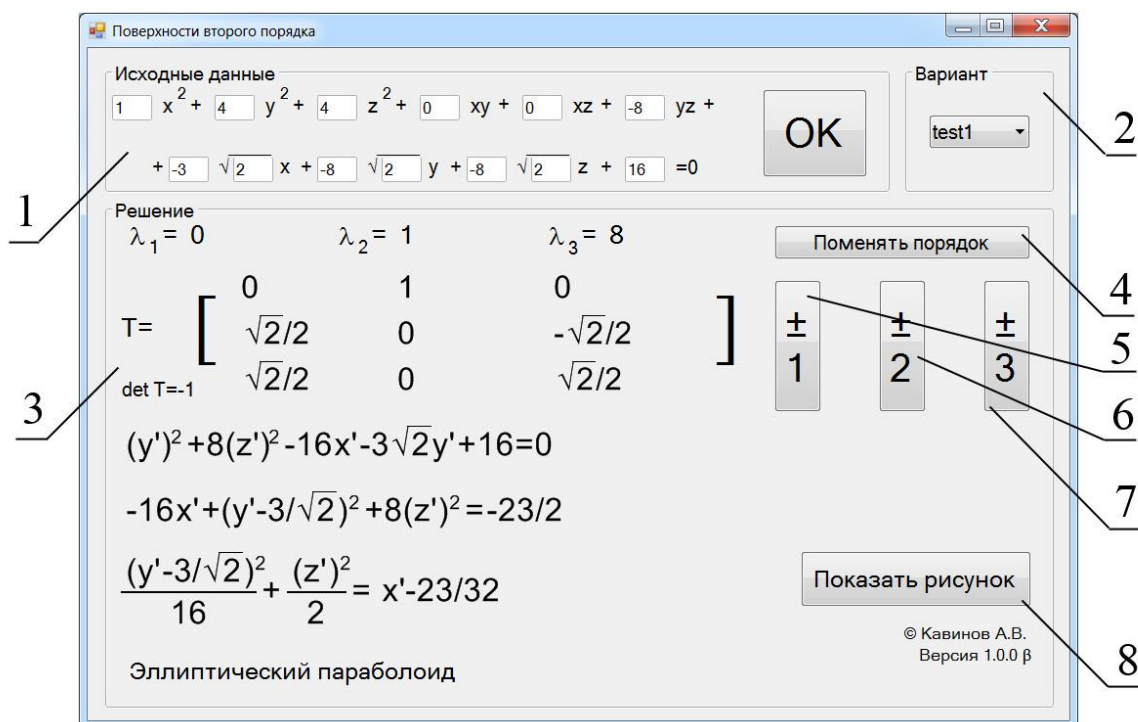


Рис. 4. Внешний вид окна программы «Поверхности второго порядка»

Элементы интерфейса в окне программы сгруппированы в области ввода данных 1, ввода варианта 2 и область решения 3. В начале работы с программой пользователь может как задать вручную исходные данные для работы программы с помощью полей ввода области 1, так и, используя область 2, выбрать один из «вшитых» в программу вариантов заданий. После выбора набора исходных данных программа выводит текстовую часть решения в область 3. Содержание выводимой в эту область информации аналогично содержанию области 3 окна программы «Кривые второго порядка». Графические построения производятся отдельной программой, о которой будет сказано далее. Интерактивность программы обеспечивается элементами пользовательского интерфейса 4-8. С помощью элемента 4 пользователь может поменять порядок собственных чисел в квадратичной форме канонического вида, с помощью элементов 5-7 - знаки собственных векторов. С помощью программы можно получить результат, соответствующий любому из сорока восьми возможных способов решения. Элемент управления 8 предназначен для вывода на экран изображения поверхности и осей канонической системы координат. При активации пользователем этого элемента вызывается третья программа комплекса, о которой будет сказано ниже.

Программа «Просмотр параметрически заданных поверхностей» позволяет проанализировать внешний вид поверхности или системы поверхностей. Она разработана при помощи бесплатной среды разработки Code Blocks, набора средств разработки

mingw-w64 и компилятора gcc. Языком разработки был выбран C++. Причиной такого выбора стала простота использования библиотеки поддержки OpenGL. В программе были также использованы собственные разработки автора в области создания пользовательского интерфейса и разработанный ранее автором компилятор математических функций. Внешний вид окна программы показан на рисунке 5.

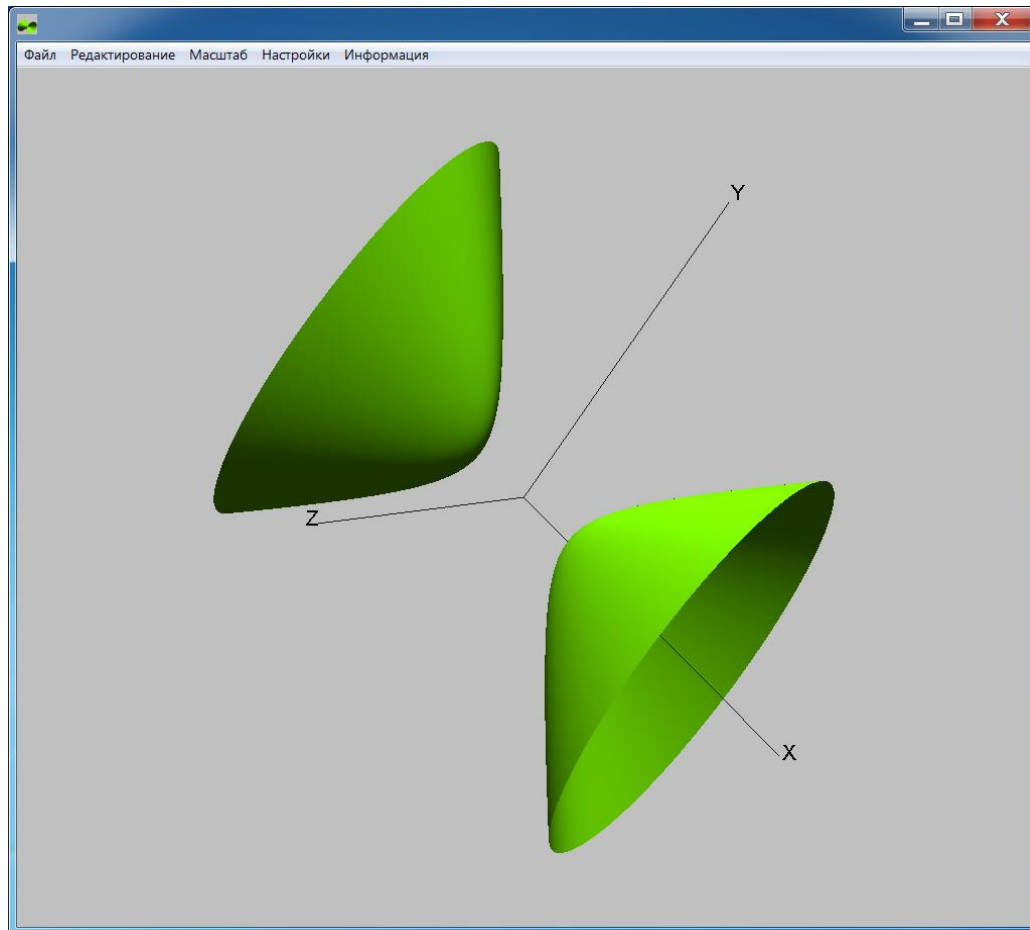


Рис. 5. Внешний вид окна программы
«Просмотр параметрически заданных поверхностей»

Программа позволяет просматривать не только поверхности второго порядка, но и любые параметрически заданные поверхности и их объединения. Изображение в окне программы может быть повернуто с помощью манипулятора «мышь», что обеспечивает возможность всестороннего анализа поверхности пользователем.

Заключение

В процессе исследования возможностей информатизации учебного процесса автором было разработано программное обеспечение для автоматизации проверки домашних заданий по дисциплине «Линейная алгебра». Использование описанных в статье программ позволило автору значительно сократить время, затрачиваемое на проверку домашних заданий, и повысить качество проверки.

Испытание на практике разработанного комплекса программ выявило высокий потенциал рассматриваемого подхода к информатизации проверки домашних заданий и показало важность аспекта обратной связи в области информатизации образования.

ССЫЛКИ НА ИСТОЧНИКИ

1. Васильева прогнозирует цифровую революцию в образовании (2017г.) URL: <https://www.pnp.ru/social/vasileva-prognoziruet-cifrovuyu-revolyuciyu-v-obrazovanii.html>
2. Девлет-Гельды Г. К. Информатизация образования как объективная закономерность развития экономики // Научно-методический электронный журнал «Концепт». – 2016. – № 5 (май). – С. 53–59. – URL: <http://e-koncept.ru/2016/16095.htm>.
3. Краснов П. С. Информационное обеспечение управления развитием образовательного учреждения // Научно-методический электронный журнал «Концепт». – 2012. – №3 (Март). – С. 56–60. – URL: <http://e-koncept.ru/2012/1228.htm>.
4. Носов А. Л. Проблемы информатизации системы образования в постиндустриальном обществе // Научно-методический электронный журнал «Концепт». – 2015. – № 1 (январь). – С. 1–5. – URL: <http://e-koncept.ru/2015/15001.htm>.
5. Лабуз Л. С., Мазаева Л. Н. Информационные технологии в высшем профессиональном образовании: проблемы и перспективы // Научно-методический электронный журнал «Концепт». – 2016. – Т. 37. – С. 90–95. – URL: <http://e-koncept.ru/2016/56791.htm>.
6. Черняева Э. П. Информационные технологии в образовательном процессе современного вуза // Научно-методический электронный журнал «Концепт». – 2016. – Т. 41. – С. 225–230. – URL: <http://e-koncept.ru/2016/56945.htm>.
7. Католиков И. В. Использование автоматизированных систем в образовательном процессе // Научно-методический электронный журнал «Концепт». – 2013. – Т. 3. – С. 1676–1680. – URL: <http://e-koncept.ru/2013/53338.htm>.
8. Норенков И.П., Зимин А.М. Информационные технологии в образовании. –М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. –352 с.
9. Информатизация образования: направления, средства, повышение квалификации / Под ред. С.И. Маслова. — М.: Изд-во МЭИ, 2004.
10. Меламуд В.Э. Информатизация образования как условие его модернизации. – М.: Московский психолого-социальный институт, 2004. – 464 с.
11. Белоус, Е.С. Преимущества и недостатки тестирования в сравнении с другими методами контроля знаний[Электронный ресурс]/ М.Э. Желнин, В.А. Кудинов, Е.С. Белоус// Ученые записки. Электронный научный журнал Курского государственного университета. –2012. –№1 (21).–Режим доступа: <http://scientific-notes.ru/pdf/023-030.pdf>
12. Морозова М.А. Компьютерное тестирование в процессе обучения инженерным специальностям: проблемы и перспективы // Педагогика. Вопросы теории и практики. – Тамбов: Грамота, 2016. № 4. С. 41-44. ISSN 2500-0039.

Kavinov Alexey Vladimirovich,

Associate Professor of the Department of FN-12 "Mathematical Modeling" of Bauman Moscow State Technical University.

alekseyvladimirovich1@yandex.ru

On informatization of checking homework on curves and surfaces of the second order

Abstract. The importance of informatization of education in modern society is difficult to overestimate. Information technology allows you to bring the learning process to a new level. In studies devoted to this problem, much attention is paid to the process of transmitting information to the trainee and much less to feedback, which makes it possible to control the process of assimilation of material by trainees. This article fills this gap, describing the author's experience in informatization of this area of the educational process. The aim of the study was to study the possibility of informatization of this kind and the effectiveness of the tools used. The author of the article has developed software whose effectiveness has been tested in practice.

Keywords: education informatization, software, linear algebra

MODERN EUROPEAN RESEARCHES: ISSUE 5 (T.1), 2019
ISSN 2311-8806

FOUNDER AND PUBLISHER
Privatuniversität Schloss Seeburg, Salzburg

EDITORIAL ADDRESS
Seeburgstrasse 8, 5201 Seekirchen am Wallersee, Salzburg, Austria
publisher@doaj.net

PRINTING HOUSE
Autonomous non-profit organization of supplementary professional education
“Inter-regional center of innovative techniques in education”
printed by permission of Privatuniversität Schloss Seeburg, Salzburg, Austria

Sent for printing 11-12-2019
Circulation 1000
Order 013117/125

© All Rights Reserved, 2019