Ярославский государственный университет
им. П. Г. Демидова

5-я научно-методическая конференция
преподавателей математического факультета
и факультета информатики
и вычислительной техники ЯрГУ

Преподавание математики и компьютерных наук в классическом университете

Материалы конференции

Ярославль,
24–25 апреля 2014 г.

УДК 378:51(063)
ББК Ч 486.24/29я43

Рекомендовано
Редакционно-издательским советом ЯрГУ
в качестве научного издания. План 2014 года

П 71 **Преподавание математики и компьютерных наук в классическом университете :** материалы конференции / отв. ред. М. В. Невский ; Яросл. гос. ун-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2014. — 168 с.

ISBN 978-5-8397-1001-6

Представлены материалы 5-й научно-методической конференции «Преподавание математики и компьютерных наук в классическом университете», состоявшейся в Ярославском государственном университете им. П. Г. Демидова в апреле 2014 г.

УДК 378:51(063)
ББК Ч 486.24/29я43

ISBN 978-5-8397-1001-6

Содержание

Предисловие

Жизнь украшается двумя вещами: занятием математикой и её преподаванием.

Симеон Дени Пуассон (1781 – 1840)

Should five per cent appear too small,
Be thankful I don't take it all.

The Beatles, «Taxman» (1966)

Палочки должны быть попендикулярны.

Вениамин Каверин, «Два капитана» (1944)

Студенты могут и утомлять, и мешать, но после сорока лет преподавания почему-то оказывается, что ученики — это самая важная часть твоей жизни. Они становятся мудрее тебя (а ты, кажется, только стареешь), они женятся, разводятся и женятся снова, они присылают фотографии своих детей и домов, они просят о рекомендательных письмах, которые вскоре образуют обширную директорию на твоём компьютере; а время от времени они поражают тебя и наполняют твоё сердце гордостью за фантастические новые теоремы и открытия, о которых ты и не мечтал.

Юрий Манин, «Математика как метафора» (2008)

Начатый несколько лет назад процесс реформирования отечественного образования набирает новые обороты. Он идёт трудно и болезненно, а многие новации весьма и весьма сомнительны. Введенный в школах единый государственный экзамен во многом превратил обучение школьников в натаскивание и тренировку по сдаче тестов. При реализации ЕГЭ выявлены не виданные ранее, чудовищные нарушения, поставившие высокие результаты экзамена под сомнение. Выпускники, выполнявшие свои задания честно, проиграли тем, кто сумел найти обходные пути. Это нанесло тяжелый удар системе воспитания учеников.

В вузовском образовании стремительно развивается так называемый компетентностный подход, хотя подходящего общепринятого определения компетенции до сих пор нет. Многие компетенции, узаконенные в федеральных государственных образовательных стандартах (ФГОС), являются чересчур общими и трудно формируемыми, в особенности за

более короткий срок обучения по данному направлению (по сравнению с прежним сроком обучения по родственной специальности). Из новых шагов реформы отметим перевод аспирантуры в уровень вузовского образования, что не способствует росту научных кадров. Большую силу в образовании набрал бюрократический подход, при котором основное движущее противоречие — между сложностью изучаемой дисциплины и возможностями обучающегося и преподавателя — вытесняется противоречием между реальной и мнимой (бумажной) компонентами процесса. Сказанное в полной мере касается и математического образования.

В Ярославском государственном университете имени П. Г. Демидова подготовка специалистов в области математики и прикладой математики осуществляется на математическом факультете и факультете информатики и вычислительной техники (ИВТ). За время своего существования факультетами подготовлено большое количество высококвалифицированных работников по ряду наукоёмких специальностей и направлений. Их выпускники входят в ведущий кадровый состав многих предприятий и организаций Ярославля, региона и всей страны, а некоторые успешно работают и за рубежом.

Опыт, накопленный в научно-методической области, преподаватели факультетов обсуждают на проводимых нами традиционных конференциях "Преподавание математики и компьютерных наук в классическом университете" и фиксируют в соответствующих сборниках материалов. История этих конференций продолжается уже десять лет. Первые четыре конференции с таким названием были проведены в 2005, 2007, 2010 и 2012 годах. Нынешняя конференция — юбилейная, пятая по счёту — состоялась в апреле 2014 года.

Организаторы конференции предложили преподавателям двух факультетов подготовить доклады следующей тематики: 1) специальные вопросы, возникающие при преподавании математических и компьютерных дисциплин; 2) вопросы методического характера, связанные с преподаванием этих дисциплин; 3) актуальные проблемы образования в области математики и компьютерных наук. В оргкомитет поступило 29 докладов преподавателей, представляющих все семь кафедр математического факультета (алгебры и математической логики; дифференциальных уравнений; компьютерной безопасности и математических методов обработки информации; математического анализа; математического моделирования; общей математики; теории функций и функционального анализа), а также четыре кафедры факультета ИВТ (вычислительных и программных систем; дискретного анализа; компьютерных сетей; теоретической информатики). Из содержания видно, что в докладах рассматриваются весьма разнообразные вопросы, связанные с преподаванием математических и компьютерных дисциплин.

Наиболее ценным представляется то, что в нынешнее неспокойное для математического образования время в докладах конференции в рамках привычной схемы *знания – умения – навыки* обсуждаются конкретные вопросы преподавания математики. Мы видим в этом наше стремление к сохранению традиций столь уважаемой специалистами старой школы отечественного математического образования.

М. Невский,
ответственный редактор,
декан математического факультета
Ярославского государственного университета им. П. Г. Демидова

УДК 378

М. А. Башкин
Ярославский государственный университет им. П. Г. Демидова
E-mail: m_bashkin@list.ru

О методе парных сравнений в курсе «Линейная алгебра»

Метод парных сравнений является одним из эффективных инструментов системного подхода к сложным проблемам принятия решений. Цель рассмотрения данного метода в курсе «Линейная алгебра» — научиться применять изученый теоретический материал к решению практических задач, приобрести навыки работы с пакетами математических программ. Студентам предлагается самостоятельно изучить метод парных сравнений и выполнить лабораторную работу, используя любой доступный пакет математических программ.

Библиография: 1 название.

Ключевые слова: линейная алгебра, метод парных сравнений, пакеты математических программ.

О методе парных сравнений (см. [1])

Метод парных сравнений позволяет ранжировать объекты, устанавливать между ними определенную иерархию. Введем шкалу относительной важности от 1 до 9, где 1 — равная важность, 9 — абсолютное превосходство. Пусть k — количество рассматриваемых критериев оценки. Определим приоритеты критериев. Каждый критерий оценивается экспертом числом κ_i $(i = 1, \ldots, k)$ по шкале относительной важности. Составим квадратную матрицу парных сравнений K порядка k, элементы которой $k_{ij} = \frac{\kappa_i}{\kappa_j}$. Эта матрица является положительно определенной, обратно-симметричной $(k_{ij} = \frac{1}{k_{ji}})$, на главной диагонали стоят 1. Следующим шагом после составления матрицы парных сравнений является вычисление вектора приоритетов. Относительная важность каждого отдельного критерия в иерархии определяется оценкой соответствующего ему элемента вектора. Метод отыскания вектора приоритетов основывается на том, что искомый вектор является собственным вектором матрицы парных сравнений, соответствующим максимальному

собственному значению. Собственное значение называется мерой согласованности критериев.

Далее, для каждого критерия строим квадратную матрицу, порядок которой равен числу оцениваемых объектов. Находим собственный вектор, соответствующий максимальному собственному значению в каждом случае. Полученные собственные векторы записываем в столбцы матрицы. Умножаем эту матрицу на собственный вектор приоритетов критериев. Полученный вектор определяет предпочтение объектов.

Об используемых пакетах прикладных программ

В настоящее время имеется довольно много разнообразных пакетов математических программ с разной степенью доступности. И практически в каждом реализован алгоритм нахождения собственных векторов и собственных значений по заданной квадратной матрице. В этой статье для выполнения вычислений используется пакет Mathematica, широко доступный с 2009 года на сайте `www.wolframalpha.com`.

Опишем используемые команды:

`Eigensystem[M]` — возвращает список собственных значений и собственных векторов квадратной матрицы `M`;

`normalize[v]` — возвращает нормированный (собственный) вектор.

Пример задания

Пусть сравниваем 3 объекта по 5 критериям. Какому объекту отдать предпочтение? Эксперт составляет матрицу критериев оценки

$$K = \begin{pmatrix} 1 & 5 & 3 & 7 & 6 \\ \frac{1}{5} & 1 & \frac{1}{3} & 5 & 3 \\ \frac{1}{3} & 3 & 1 & 6 & 3 \\ \frac{1}{7} & \frac{1}{5} & \frac{1}{6} & 1 & \frac{1}{3} \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 3 & 1 \end{pmatrix}.$$

Находим собственные векторы и собственные значения. Выбираем наибольшее собственное значение и соответствующий ему вектор:
$\lambda_{max} = 5.29941$, $X = (0.863094;\ 0.241011;\ 0.418885;\ 0.0662734;\ 0.130863)^T$.

Итак, самый важный критерий — первый; самый малозначительный — четвертый.

По той же шкале эксперт оценивает три объекта по каждому из пяти критериев:

$$1\text{:}\ \begin{pmatrix} 1 & 6 & 8 \\ \frac{1}{6} & 1 & 4 \\ \frac{1}{8} & \frac{1}{4} & 1 \end{pmatrix}, \qquad 2\text{:}\ \begin{pmatrix} 1 & 7 & \frac{1}{5} \\ \frac{1}{7} & 1 & \frac{1}{8} \\ 5 & 8 & 1 \end{pmatrix}, \qquad 3\text{:}\ \begin{pmatrix} 1 & 8 & 6 \\ \frac{1}{8} & 1 & \frac{1}{4} \\ \frac{1}{6} & 4 & 1 \end{pmatrix},$$

$$4\text{:}\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \qquad 5\text{:}\ \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix}.$$

В каждом из 5 случаев находим собственные векторы и собственные значения. Выбираем для каждого случая наибольшее собственное значение и соответствующий ему вектор:

1: $\lambda_{max1} = 3.13561,\ Y_1 = (0.96887;\ 0.232892;\ 0.0839721)$;
2: $\lambda_{max2} = 3.24695,\ Y_2 = (0.31008;\ 0.0724493;\ 0.947946)$;
3: $\lambda_{max3} = 3.13561,\ Y_3 = (0.96887;\ 0.083972;\ 0.232892)$;
4: $\lambda_{max4} = 3.00000,\ Y_4 = (0.57735;\ 0.57735;\ 0.57735)$;
5: $\lambda_{max5} = 3.01829,\ Y_5 = (0.276186;\ 0.632309;\ 0.723814)$.

Составляем матрицу, записывая найденные векторы $Y_1, Y_2, \ldots, Y_5$ в столбцы, и умножаем ее на X. Получаем:

$$\begin{pmatrix} 0.96887 & 0.31008 & 0.96887 & 0.57735 & 0.276186 \\ 0.232892 & 0.0724493 & 0.083972 & 0.57735 & 0.632309 \\ 0.0839721 & 0.947946 & 0.232892 & 0.57735 & 0.723814 \end{pmatrix} \cdot \begin{pmatrix} 0.863094 \\ 0.241011 \\ 0.418885 \\ 0.0662734 \\ 0.130863 \end{pmatrix} =$$

$$= \begin{pmatrix} 1.39121 \\ 0.374652 \\ 0.53148 \end{pmatrix}$$

Следовательно, предпочтение косвенным образом нужно отдать первому объекту.

Ссылки

1. *Саати Т.* Принятие решений: Метод анализа иерархий/ пер. с англ. М. : Радио и связь, 1993. 278 с.

УДК 510.21

Л. Ю. Белова, Ю. А. Белов
Ярославский государственный университет им. П. Г. Демидова
E-mail: luk1945@yandex.ru
E-mail: belov45@yandex.ru

Роль курсов дискретной математики и математической логики в базовом образовании математиков-прикладников

В заметке намечена тема роли указанных в заголовке учебных курсов в структуре базового образования математиков-прикладников и специалистов по фундаментальной информатике.Затрагиваются и вопросы содержания данных курсов.

Библиография: 1 название.

Ключевые слова: дискретная математика, преподавание.

В действительности, вопросы, которые необходимо обсудить в связи с указанной темой, очень серьёзны и велики. В принципе, основой предлагаемых к рассмотрению вопросов должно быть, видимо, чёткое осознание того, чем же существенно различаются «профили» или модели специалистов математиков и математиков-прикладников или специалистов по информатике. По нашему мнению, основой образования прикладников, в отличие от«чистых математиков», во многом является теоретическая информатика (что следует даже из названия специальности), и это самое существенное различие этих специалистов. Из признания этого утверждения следуют многие практические следствия по структуре, организации и объёмам рассматриваемых курсов в сравнении с другими. Нечто подобное уже высказывалось не раз, например в [1], однако никаких позитивных сдвигов не происходит. В нашем ближайшем кругу весьма популярна критика руководства, особенно высшего. Однако данный вопрос высшим руководством как раз оставлен на наше усмотрение, и вот здесь положение весьма плачевно.

Заметим, указанные курсы рассматриваются нами совместно, так как они близки генетически и вообще до недавнего времени составляли

один курс. Кратко наши предложения таковы: существенно увеличить объём указанных курсов, возможно, за счёт курсов анализа и других аналитических курсов. Очевидно, данные курсы должны составлять теоретическую основу для таких учебных дисциплин, как теория алгоритмов, теория автоматов и машин Тьюринга-Поста, теория доказательств, теория формальных языков и грамматик, методы трансляции, языки программирования, теория кодирования, теория распределённых информационных систем, теория транспортных протоколов и других. Для выполнения этой задачи – быть основой существенной части образования специалиста – требуется большой, серьёзно продуманный по содержанию курс, сопровождаемый примыкающими к нему специальными дисциплинами.

Предлагается освободить курс дискретной математики от такой «общественной нагрузки», как изложение элементов теории множеств, а также от подробного изучения различных алгоритмов – на графах, перестановках, булевых фнукциях и т. п. Подобные вопросы должны входить в различные специальные дисциплины, примыкающие по содержанию к основному курсу. Курс дискретной математики должен стать в полной мере базовым теоретическим курсом, основным содержанием которого должны быть структурные, классификацинные теоремы, а не алгоритмы.

Курс логики нельзя ограничить лишь изложением исчисления высказываний и теории булевых функций. Кроме исчисления предикатов, необходимо дать какие-то элементы формальных систем, теорий вычислимости и разрешимости. Без этого невозможно содержательно изложить многие из упомянутых выше дисциплин, составляющих основу образования настоящего специалиста по **фундаментальной информатике и информационным технологиям**, как указывается в новом направлении бакалавриата.

Ссылки

1. *Белов Ю. А.* О содержании курса «Дискретная математика» в университете. // Математика и математическое образование : Теория и практика. Вып. 3. Ярославль : ЯГТУ, 2002. С. 34–37.

УДК 51:378

Ю. И. Большаков
Ярославский государственный университет им. П. Г. Демидова
E-mail: bolsh@uniyar.ac.ru

Иллюстрация действия групп на множествах и некоторые задачи классификации матриц

Настоящая заметка тесно связана с опубликованной автором статьей в материалах 4-й научно-методической конференции преподавателей математического факультета и факультета ИВТ ЯрГУ им. П. Г. Демидова. Предлагаемые в настоящей работе примеры служат иллюстрацией действия групп на множествах в различных задачах, связанных с классификациями соответствующих объектов множества.

Библиография: 6 названий.

Ключевые слова: конечные множества, счетные множества, континуальные множества, отношение эквивалентности, задачи классфикации матриц.

Понятие действия группы G на множестве X, по всей вероятности, возникло ранее понятия абстрактной группы. Причиной, по которой изучалась пара $(G,\ X)$, а не группа G в отдельности, служило существовавшее тогда представление об окружающем мире. Считалось, что мир устроен по Евклиду и часто «доказательства» опирались на наглядные представления о движении объектов в пространстве; так, например, в книге [1, с. 20] говорится: «Равными фигурами называются такие две фигуры, которые можно совместить одну с другой так, чтобы они в точности совпадали во всех своих частях».

Эта теория принята, фактически, и в современной математике: действие группы G на множестве X рассматривается как гомоморфизм из G в $Is(X)$ — группу всех биекций X на себя [2, с. 301]. Строго говоря, действие G на X определяется неоднозначно, ибо гомоморфизмов $\varphi_i : G \to Is(X)$ может существовать достаточно много, многое зависит от мощностей G и X. Если i задано, то мы можем символ φ_i опускать и вместо $\varphi_i(g)(x)$ писать просто $g(x),\ g \in G,\ x \in X$. Подобный

теоретико-групповой подход весьма полезен при изучении студентами-математиками различных курсов алгебраического и геометрического характера.

Итак, пусть X — произвольное множество, $Is(X)$ — группа (относительно операции композиции отображений), состоящая из всех биекций X на себя. Мы будем говорить, что группа G действует на X с помощью F, если F — гомоморфизм групп $F : G \to Is(X)$.

Если F инъективно, то говорят, что G действует на X (с помощью F) эффективно. Фактически, это означает, что F осуществляет изоморфное вложение G в $Im\ F \subset Is(X)$. Так, например, для любой подгруппы $G \subset GL(n, \mathbb{R})$ её действие на $X = \mathbb{R}^n$, заданное с помощью F формулой $F(g)(x) = g \cdot x$, эффективно. Здесь $g \in G,\ x \in X, \cdot$ — операция матричного умножения. На этом простом примере легко убедиться, что F, действительно, задает действие группы G на X, т. е. что $F(g)$ — биекция X на X и F — гомоморфизм групп G и $Is(X)$, поскольку операция матричного умножения ассоциативна. Эффективность же следует из того, что матричное тождество $h \cdot x = 0\ \ (h \in G,\ x \in \mathbb{R}^n)$ приводит к равенству $h = 0$.

Мы будем говорить, что G действует на X транзитивно, если для любой пары $(x, y) \in X^2$ существует элемент $g \in G$ такой, что $F(g)(x) = y$. Множество X при этом называется однородным пространством относительно G. В вышерассмотренном примере действие G на $X = \mathbb{R}^n$ транзитивным не будет, ибо $F(g)(0) = 0$, но то же самое действие $G = GL(n, \mathbb{R})$ на $Y = \mathbb{R}^n \setminus \{0\}$ уже транзитивно, и, соответственно, Y — однородное пространство относительно G. Назовем подмножество $O(x_0) := \{y \in X | y = F(g)(x_0),\ g \in G\}$ орбитой элемента $x_0 \in X$.

Ясно, что $O(x_0)$ — максимальное (по включению) подмножество в X, содержащее точку x_0, на котором группа G действует транзитивно. Более того, множество X — объединение непересекающихся орбит.

Так, в двух вышерассмотренных примерах с группой $G = GL(n, \mathbb{R})$ множество $X = \mathbb{R}^n = O(e_1) \cup O(0)$ состоит из двух орбит ($e_1 = (1, 0,\ \ldots,\ 0)^t$, $O = (0,\ \ldots,\ 0)^t$) а множество $Y = \mathbb{R}^n \setminus \{0\}$ — из одной относительно группы $G = GL(n, \mathbb{R})$.

Из вышеприведенных соображений следует, что задача перечисления орбит равносильна задаче классификации по следующему отношению эквивалентности: $x \sim y \Leftrightarrow \exists g \in G | y = F(g)(x)$. Для ее решения (в той или иной форме) достаточно найти подмножество $\mathbf{B} \subset X$, такое что $\bigcup_{b \in \mathbf{B}} O(b) = X$ и $O(b_1) \bigcap O(b_2)$, либо $\emptyset$, либо $b_1 = b_2$. В последнем случае очевидно, что $O(b_1) = O(b_2)$. При таком определении множество $\mathbf{B}$ состоит из всех неэквивалентных в X элементов. Так, например, если $G = GL(n, \mathbb{R}),\ X = \mathbb{R}^n,\ \mathbf{B} = \{0,\ (1,\ 0, \cdots, 0)^t\},\ card\mathbf{B} = 2$, если $G = GL(n,\ \mathbb{R}),\ X = \mathbb{R}^n \setminus \{0\}$; то $\mathbf{B} = \{(1,\ 0, \cdots, 0)^t\},\ card\mathbf{B} = 1$ (т. е. G действует транзитивно на X).

Рассмотрим несколько нетривиальных примеров действия группы G на X.

Пример 1 [3, с. 11]. Пусть $G = \mathbb{Z},\ X = [0,\ 1]$. Действие G на X определено по формуле: $F(m)(a) = a^{2^m},\ a \in X,\ m \in \mathbb{Z}$. Множество $\mathbf{B} = \{0; 1; [p^2, p)\}$, здесь $p \in (0; 1)$ — любой фиксированный элемент; орбиты $O(0) = 0;\ O(1) = 1;\ O(p) \subset (0; 1)$ и $cardO(p) = card\mathbb{Z}$.

$\triangle$ 1. Покажем, что $ImF(m) \subset X$ при всех $m \in \mathbb{Z}$. Для 0 и 1 это очевидно; если же $a \in (0; 1)$, т. е. $0 < a < 1$, то поскольку a^x убывает от 1 до 0 с возрастанием x от 0 до $+\infty$, то $Ima^x \subset (0; 1)$. Полагая $x = 2^m$, получим требуемое включение.

2. $F(m)$ – биекция X на X. Инъективность $F(m)$: если $p^{2^m} = q^{2^m}$, то, возведя обе части равенства в степень 2^{-m}, получим $p = q$. Сюръективность $F(m)$: если $0 < b < 1$, то равенство $p^{2^m} = b$ равносильно $p = b^{2^{-m}}$ и, очевидно, $0 < p < 1$. Если $b = 0$ или $b = 1$, то параметр $p = 0$ или $p = 1$ соответственно.

3. Гомоморфность F : $F(m + n) = F(m) \circ F(n) \iff \forall a \in X$ $F(m + n)a = F(m)(F(n))a \Leftrightarrow a^{2^{m+n}} = (a^{2^n})^{2^m} = a^{2^n \cdot 2^m} = a^{2^{m+n}}$.

4. Пусть $0 < p < 1$. Покажем, что $[p^2,\ p)$ состоит из неэквивалентных элементов. Если $b = a^{2^m}$ и $0 < p^2 \le a < b < p < 1$, т. е. $0 < p^2 \le a < a^{2^m} < p < 1 \Leftrightarrow +\infty > 2\log_a p \ge 1 > 2^m > \log_a p > 0 \Leftrightarrow +\infty > 1 + \log_2 log_a p \ge 0 > m > \log_2 \log_a p > -\infty$. Пусть $z = \log_2 \log_a p$, тогда $1 + z \ge 0 > m > z \Rightarrow z + 1 > m > z$. Такого m не существует, ибо $m \in \mathbb{Z}$.

Пусть $0 < b < 1$, тогда $\exists! m \in \mathbb{Z}$ такой, что $p^2 \le b^{2^m} < p$. Последнее неравенство равносильно неравенству $2 \ge \log_p b^{2^m} > 1 \Leftrightarrow 2 \ge 2^m \log_p b > 1 \Leftrightarrow 1 \ge m + \log_2 log_p b > 0$. Пусть $log_2 log_p b = c$, тогда $1 \ge m + c > 0 \Leftrightarrow -c < m \le 1 - c$. Ясно, что если m существует, то оно единственно; положим $m = [-c] + 1$, тогда последнее равенство равносильно $-c < [-c] + 1 \le 1 - c$ и поскольку $-c = [-c] + \{-c\}$, то $-c < -c - \{-c\} + 1 \le 1 - c \Leftrightarrow 0 < -\{-c\} + 1 \le 1 \Leftrightarrow \{ \begin{matrix} \{-c\} < 1 \\ \{-c\} \ge 0 \end{matrix}$ верно. Таким образом, $m = 1 + [-\log_2 \log_p b] = 1 + [\log_2 \log_b p]$. Поскольку точки 0 и 1 остаются на месте, то $\mathbf{B} = \{0; 1; [p^2, p)\}$, где p — любой фиксированный элемент из интервала $(0; 1)$. Заметим, что действие $\mathbb{Z}$ на X не будет транзитивным, ибо орбит континуум, но будет эффективным.

В каждом из нижеперечисленных примеров свойства 1–4 для функции F (как в примере 1) могут быть проверены непосредственно, но мы этого делать не будем.

Пример 2. $G = GL(n, \mathbb{R}),\ X = \mathbb{R}^{n \times n},\ F(g)(x) = gxg^{-1}$. При таком определении F множество $\mathbf{B}$ состоит из всевозможных вещественных жордановых форм (см. [4, с. 181 — 186]) матрицы $x \in \mathbb{R}^{n \times n}$ с точностью до перестановки жордановых блоков. Заменяя g на λg $(\lambda \ne 0)$, мы видим, что действие группы $GL(n, \mathbb{R})$ эффективным не будет.

Пример 3. $G = GL(n,\ \mathbb{R}), X = \{x \in \mathbb{R}^{n\times n} | x^t = x\}$. $F(g)x = gxg^t$. Используя канонический вид матрицы квадратичной формы над $\mathbb{R}$ [5, с. 174 – 179], мы находим, что $\mathbf{B} = \{x \in X | x = diag(1, \ldots, 1, -1, 0, \ldots, 0)$ с условием $\pi+\mu+\nu = n$, где π — число 1, μ — число -1, ν — число нулей в матрице $x \in \mathbf{B}$. Поскольку на тройку $(\pi,\ \mu,\ \nu)$ условие $\pi + \mu + \nu = n$ единственное, то множество $\mathbf{B}$ конечно и $card\ \mathbf{B} = \frac{1}{2}(n + 1)(n + 2)$.

Пример 4. $G = GL(n,\ \mathbb{R}) \times GL(n, \mathbb{R}),\ X = \mathbb{R}^{n\times n};\ F(g,\ h)(x) = gxh^{-1}$. Поскольку такого сорта отображение служит критерием сохранения ранга матрицы x [6, с. 70–73], то $\mathbf{B} = \{x \in X | x = diag\ (1,\ \ldots,\ 1, 0,\ \ldots, 0)\}$ с единственным условием на число единиц равным рангу матрицы x, поэтому множество $\mathbf{B}$ конечно и $card\ \mathbf{B} = n + 1$.

В заключение рассмотрим 2 примера геометрического характера.

Пример 5. $G = O(2) = \{g \in GL(2, \mathbb{R}) | g^t g = I\},\ X = \mathbb{R}^2$ – евклидова плоскость; $F(g)(x) = g \cdot x$. Легко видеть, что $O(x_0) = \omega(0,\ ||x_0||)$, где O – центр окружности ω, $||x_0||$ – её радиус. Поэтому множество $\mathbf{B}$ — луч, исходящий из точки O в направлении вектора $e_1 = (1; 0)^t$.

Пример 6. $G = SL(2, \mathbb{R}) \times \mathbb{R}^2,\ X$ – множество всех параллелограммов, расположенных на евклидовой плоскости $\mathbb{R}^2$. Здесь $g = (A,\ \xi),\ A \in GL(2,\ \mathbb{R}),\ det\ A = 1; \xi \in \mathbb{R}^2$. Действие группы G на X задается формулой: $F(g)(x) = F(A,\ \xi)(x) := Ax + \xi$; умножение $A \cdot x$ поточечное (параллелограмм переводит в параллелограмм, то же самое делает и сдвиг).

Поэтому множество $\mathbf{B}$ — множество всех квадратов, расположенных в первой четверти плоскости $\mathbb{R}^2$ с общей вершиной в начале координат 0.

В заключение отметим, что множество $\mathbf{B}$ в каждом из примеров определено неоднозначно. Каждую его точку можно «подкрутить» элементом группы, но мощность $\mathbf{B}$ при этом сохранится.

Ссылки

[1] *Адамар Ж.* Элементарная геометрия. Ч. I. М. : Учпедгиз, 1958. 608 с.

[2] *Кострикин А. И.* Введение в алгебру. М. : Наука, 1977. 496 с.

[3] *Большаков Ю. И.* Теоретико-групповой подход в рамках курса НОШКМ // Преподавание математики и компьютерных наук в классическом университете: материалы 4-й научно-методической конференции преподавателей мат. ф-та и ф-та ИВТ ЯрГУ им. П. Г. Демидова. Ярославль, 2012. С. 9 – 12.

[4] *Шилов Г.Е.* Математический анализ: Конечномерные линейные пространства М. : Наука, 1969. 432 С.

[5] *Курош А. Г.* Курс высшей алгебры. М. : Наука, 1968. 432 с.

[6] *Гантмахер Ф. Р.* Теория матриц. 4-е изд. М. : Наука, 1988. 552 с.

УДК 519.681.5: 519.682

В. В. Васильчиков
Ярославский государственный университет им. П. Г. Демидова
E-mail: vasilch@uniyar.ac.ru

Компоненты для организации параллельных вычислений в .NET Framework

Рассматриваются библиотеки классов для разработки рекурсивно-параллельных программ в .NET Framework. Они могут быть использованы при изучении курсов по параллельному программированию, а также при подготовке курсовых и дипломных проектов.

Библиография: 3 названия.

Ключевые слова: параллельные вычисления, .NET Framework.

Состав и назначение компонентов

Автором были разработаны два основных компонента [1, 2] для организации рекурсивно-параллельных (РП) вычислений в локальной сети с поддержкой .NET Framework версии 4.0 или более новой: коммуникационный модуль и собственно библиотека поддержки РП-программирования.

Каждый из них представляет собой динамически подключаемую библиотеку, которая может использоваться в программах для .NET Framework версии 4.0 с любым типом интерфейса (консольный, WPF или основанный на Windows Forms). Пользователю они предоставляются в виде DLL-файлов, которые он должен включить в свой проект. Предполагается, что при разработке программы используется среда Visual Studio 2010 или более новая. Разумеется, в этом случае разработчик может воспользоваться всеми возможностями, предоставляемыми IntelliSense. Кроме того, для обоих компонентов дополнительно предоставляется развитый файл контекстной справки в формате CHM.

Коммуникационный модуль

Модуль предоставляется в виде файла CommModule.dll (файл справки CommModule.chm). Основными функциональными возможностями компонента являются следующие.

При запуске пользователю (независимо от типа основного интерфейса приложения) предоставляется удобный WPF-интерфейс для связи по протоколу TCP приложений, запущенных на разных компьютерах локальной сети, по принципу «каждый с каждым».

При установлении связи не требуется вводить ни имена, ни IP-адреса — на первом этапе соединения используется широковещание по протоколу UDP. Поскольку упомянутый протокол не гарантирует доставку сообщений, предусмотрены средства для посылки дополнительных сообщений, побуждающих участников процесса соединения выполнить необходимые действия. После завершения этапа первоначального установления связи каждый компьютер сети связан с каждым по протоколу TCP, каждому присвоен номер для его идентификации, и управление передается приложению-клиенту.

Для удобства отладки на одном компьютере предусмотрена возможность запускать несколько (в настоящей версии до четырех, автор полагает, что этого достаточно) экземпляров приложения, они будут связываться друг с другом так же, как если бы работали на разных рабочих станциях. В этом случае можно даже автоматически удобно расположить на экране окна разных копий приложения, реализующих выбранный разработчиком пользовательский интерфейс (в том числе, и консольные окна).

Классы коммуникационного модуля предоставляют разработчику удобные методы для надежной передачи на удаленный компьютер любого сериализуемого объекта и обеспечивают восстановление там его состояния. Они также предлагают средства для пересылки по сети пользовательских сообщений и организации их обработки по мере приема. При этом все необходимые действия по синхронизации, преобразованию сложных объектов в поток байтов и восстановлению их на удаленном компьютере осуществляются автоматически.

Область применения предложенного коммуникационного модуля не ограничивается его использованием со стороны библиотеки поддержки рекурсивно-параллельного стиля программирования, хотя изначально он предназначался именно для этого. Компонент может быть полезен в любых сетевых программах для .NET Framework, поскольку позволяет очень легко установить полносвязное соединение и передавать объекты любого уровня сложности, не заботясь о множестве технических деталей.

Библиотека поддержки рекурсивно-параллельного программирования

Компонент предоставляется в виде файла RPM_ParLib.dll (файл справки RPM_ParLib.chm). В своей работе он использует коммуникационный модуль CommModule.dll, который должен находиться в той же директории. Ниже описываются основные функциональные возможности библиотеки. Эти возможности, а также механизмы их реализации практически не отличаются от описанных в [3], однако в предлагаемой версии библиотеки в качестве языка программирования выступает не C, а C# (точнее, любой язык программирования для .NET Framework). Поэтому способ доступа к предлагаемым методам основан на принципах объектно-ориентированного программирования (ООП) и типичен для любых ООП-языков.

При запуске приложения инициируется установление сетевого соединения приложений, запущенных на разных компьютерах локальной сети, по принципу «каждый с каждым». Для сетевого взаимодействия библиотека использует упомянутый выше коммуникационный модуль.

После установления начального соединения пользователь может выбрать «главный» процессорный модуль (ПМ), в качестве которого может выступать один из запущенных экземпляров приложения (неважно, на разных рабочих станциях они запущены или на одной). «Главным» он называется только потому, что первым начинает вычисления, управление всей последующей работой носит децентрализованный характер.

Весь процесс вычислений должен быть оформлен как рекурсивный метод либо несколько методов, допускающих рекурсивный вызов. Поскольку в C# нет глобальных методов, в качестве «обертки» для своего рекурсивного метода разработчик должен объявить класс-наследник библиотечного класса ParMethodBase и разместить необходимый код в переписанном (override) методе ParMethod(). Последний принимает в качестве единственного параметра экземпляр класса-наследника библиотечного класса ParamBase, который служит и для передачи рекурсивному методу необходимой информации, и для возврата результатов. Собственно, все требования к оформлению кода сводятся к использованию упомянутых двух библиотечных классов. Вся предлагаемая разработчику функциональность реализуется через вызов их методов.

Библиотека предоставляет удобные методы для разбиения процесса вычислений на параллельные ветви с использованием рекурсии, автоматическое начальное распределение работы по системе, динамическую балансировку загрузки в процессе вычислений, возврат результатов с удаленных модулей, простую (по крайней мере, для разработчика прикладной программы) синхронизацию. Разумеется, при написании кода

необходимо понимать основные особенности поведения параллельной программы, в частности важность правильной синхронизации отдельных ветвей и правильной организации работы с общими данными. Для более тонкой настройки механизмов распределения работы предусмотрены специальные методы и свойства, однако на начальных этапах работы вполне подойдут и их значения по умолчанию.

Так же, как и в [3], в рассматриваемой версии библиотеки предусмотрены два класса памяти для работы с общими данными: данные, копия которых присутствует на каждом ПМ, а также общие данные, которые распределены по отдельным модулям системы, однако могут потребоваться любому ПМ. Эти данные размещаются во время работы программы динамически, предусмотрено несколько методов для доступа к ним как по чтению, так и по записи с различной семантикой запроса. Однако для эффективной работы параллельной программы следует тщательно продумать способ размещения данных и способ доступа к ним, поскольку передача данных по сети сопряжена со значительными временными задержками.

Использование в учебном процессе

Автор предполагает, что разработанное им программное обеспечение может использоваться не только при программировании трудоемких вычислений для решения научных задач, но и в учебном процессе при изучении основ параллельного программирования и при подготовке курсовых и дипломных проектов. В качестве преимуществ такого его использования можно отметить следующие.

Разработчик конечного приложения не должен заботиться о способе и механизме распределения работы, при создании исходного кода даже не требуется знать количество и характеристики компьютеров, образующих сеть для вычислений, он инвариантен к этим характеристикам. От программиста требуется только породить достаточное количество активаций параллельной процедуры, об остальном позаботится библиотека. При этом не исключается возможность ручного разбиения и распределения работы по системе. Правда, программист должен все-таки правильно понимать, как организовать это разбиение и что такое "достаточное количество". Этот вопрос, в частности, рассматривается в следующем докладе автора.

Вопросы синхронизации вычислений и использования общих данных при разработке параллельных приложений очень непросты (в особенности для начинающих), а поиск и устранение ошибок, вызванных неправильной синхронизацией, весьма не тривиален даже с использованием современных средств отладки. Парадигма РП-программирования предлагает очень простую модель синхронизации, понятную даже но-

вичку. При этом механизмы обеспечения правильной работы этой «простой модели» вовсе не так просты, но это уже не забота прикладного программиста.

Не всегда при разработке программы имеется возможность делать это на реальной сети, особенно на начальных этапах написания кода и отладки. Как уже было сказано выше, в библиотеках реализована возможность имитации работы в сети путем запуска нескольких экземпляров приложения на одном компьютере. Это существенно облегчает отладку, поскольку отладчик Microsoft Visual Studio можно «на лету» подцепить к любому выполняющемуся процессу.

Все сказанное выше позволяет сделать вывод, что рассматриваемые программные средства могут использоваться даже на начальных этапах изучения параллельного программирования. От студента требуется только знание основ разработки приложений на любом языке программирования для .NET Framework.

Ссылки

1. *Васильчиков В. В.* Коммуникационный модуль для организации полносвязного соединения компьютеров в локальной сети с использованием .NET Framework : Свидетельство о государственной регистрации программы для ЭВМ № 2013619925, 2013.

2. *Васильчиков В. В.* Библиотека поддержки рекурсивно-параллельного программирования для .NET Framework : Свидетельство о государственной регистрации программы для ЭВМ № 2013619926, 2013.

3. *Васильчиков В. В.* Средства параллельного программирования для вычислительных систем с динамической балансировкой загрузки. Ярославль : ЯрГУ, 2001.

УДК 519.681.5: 519.682

В. В. Васильчиков
Ярославский государственный университет им. П. Г. Демидова
E-mail: vasilch@uniyar.ac.ru

О распараллеливании метода Брона-Кербоша

Рассматриваются различные подходы к распараллеливанию алгоритма Брона-Кербоша для нахождения максимальной клики в неориентированном графе. На их примере демонстрируются некоторые факторы, оказывающие решающее влияние на эффективность рекурсивно-параллельных алгоритмов.

Библиография: 5 названий.

Ключевые слова: параллельные вычисления, .NET Framework.

Задача о клике и последовательный алгоритм ее решения

Напомним, что кликой называется любой полный подграф неориентированного графа (т. е. подграф, в котором каждая вершина соединена с каждой). Задача состоит в нахождении клики максимального размера.

Как известно, задача о клике относится к так называемым NP-полным задачам, для которых на данный момент неизвестно алгоритма решения с полиномиальной трудоемкостью. Так как более быстрого варианта нахождения решения нет, то остается лишь один способ – перебор, разумеется, не полный, а оптимизированный. Одним из лучших на сегодняшний день алгоритмов для решения задачи о клике считается алгоритм Брона-Кербоша (вариант метода ветвей и границ) [1].

Данный алгоритм оперирует тремя множествами вершин графа:

- Множество *compsub* – множество, содержащее на каждом шаге рекурсии полный подграф для данного шага. Строится рекурсивно.

- Множество *candidates* – множество вершин, которые могут увеличить *compsub*.
- Множество *not* – множество вершин, которые уже использовались для расширения *compsub* на предыдущих шагах алгоритма.

Алгоритм является рекурсивной процедурой, применяемой к этим трем множествам:

ПРОЦЕДУРА $bron\_kerbosh(compsub, candidates, not)$
0 ПОКА *candidates* НЕ пусто И *not* НЕ содержит вершины,
СОЕДИНЕННОЙ СО ВСЕМИ вершинами из *candidates*
ВЫПОЛНЯТЬ
1 Выбираем вершину v из *candidates* и добавляем её в *compsub*
2 Формируем $new\_candidates$ и new_not, удаляя из *candidates* и *not* вершины, НЕ СОЕДИНЕННЫЕ с v
3 ЕСЛИ $new\_candidates$ и new_not пусты
4 ТО *compsub* – клика
5 ИНАЧЕ
рекурсивно вызываем
$bron\_kerbosh(compsub, new\_candidates, new\_not)$
6 Удаляем вершину v из *compsub* и *candidates*, добавляем ее в *not*

Вариант распараллеливания №1

Для обеспечения возможности распараллеливания алгоритм решения задачи можно переформулировать следующим образом:

ПРОЦЕДУРА $par\_bron\_kerbosh(compsub, candidates, not, bound)$
0 ЕСЛИ *not* содержит вершину, СОЕДИНЕННУЮ СО ВСЕМИ вершинами из *candidates*
ТО завершить процедуру
1 ЕСЛИ размер *candidates* не превышает *bound*
ТО вызываем последовательный вариант
$bron\_kerbosh(compsub, candidates, not)$
завершаем процедуру
2 Выбираем первую вершину v из *candidates*

3 Формируем $new\_candidates$ и new_not, удаляя из $candidates$ и not вершины, НЕ СОЕДИНЕННЫЕ с v
4 ЕСЛИ $new\_candidates$ и new_not пусты
ТО $compsub \cup v$ – клика
5 ИНАЧЕ вызываем
$par\_bron\_kerbosh(compsub \cup v, new\_candidates, new\_not, bound)$
6 Удаляем вершину v из $compsub$ и $candidates$, добавляем ее в not
7 Рекурсивно вызываем $par\_bron\_kerbosh(compsub, candidates, not, bound)$

Здесь дополнительно фигурирует параметр $bound$, задающий предельный размер списка кандидатов для дальнейшего разбиения задачи на параллельные ветви (такое ограничение нужно для сокращения накладных расходов на распараллеливание [2]). Очевидно, что вызовы из пунктов 5 и 7 могут выполняться параллельно. Здесь для компактности изложения опущена проверка перспективности оставшихся на ветви вычислений (все-таки это метод ветвей и границ).

В [3] для тестирования версии РП-библиотеки для языка C++ алгоритм выстраивался примерно так же. Там же было отмечено, что трудоемкость вычислений на параллельных ветвях 5 (назовем ее «левой») и 7 («правая» ветвь) заметно различается. При проведении описанного там эксперимента эту особенность можно было даже считать полезной, поскольку она позволяла оценить качество работы распределительного механизма, что, кстати, и было основной целью.

Вариант распараллеливания №2

Для тестирования библиотек поддержки РП-программирования для .NET Framework автором была взята в качестве модельной та же самая задача и реализован параллельный алгоритм ее решения на языке C# с использованием компонентов [4] и [5]. Для тестирования генерировался случайный граф с заданным количеством вершин N и вероятности наличия ребра p. Очень быстро стало ясно, что достигаемое ускорение не соответствует ожиданиям – на графе с $N = 100$, $p = 0.5$ оно даже на сети с 10 рабочими станциями едва превышало 2. Как выяснилось, несмотря на естественность рассмотренного выше варианта рекурсивного распараллеливания, он обладает существенным недостатком.

При сравнительно небольшой плотности графа трудоемкости «левой» и «правой» ветвей различаются настолько сильно, что разбиение задачи на две ветки приводит к «отщипыванию» очень небольших кусочков вычислений, так что механизм динамической балансировки просто лишен возможности равномерно распределить работу. В эксперименте же, описанном в [3], вероятность ребра бралась равной 0.9, по-

этому при не слишком большом количестве исполнителей работу можно было распределить более или менее равномерно.

В качестве альтернативного способа построения параллельного алгоритма рассмотрим следующий. Пусть $SubTask_n$ – подмножество всех вариантов, построенное для следующих исходных значений трех основных подмножеств алгоритма Брона-Кербоша:

- Множество *compsub* состоит из одной вершины n.
- Множество *candidates* состоит из вершин, смежных с n.
- Множество *not* состоит из вершин с номерами от 0 до $n-1$ (при программировании удобнее использовать нумерацию от нуля).

Очевидно, что перебирать варианты $SubTask_n$, где n лежит в пределах от 0 до $N-1$, можно независимо, причем пересечение этих множеств пусто, а объединение содержит в себе все возможные клики. Для поиска решения на каждом множестве $SubTask_n$ вполне годится приведенный выше последовательный алгоритм Брона-Кербоша. Можно было бы и здесь применить распараллеливание, но смысла, по-видимому, в этом нет, по крайней мере, в случае, если N хотя бы вдвое больше количества имеющихся процессорных модулей, что верно фактически всегда, когда вообще есть смысл задействовать параллелизм.

Теперь рекурсивно-параллельный алгоритм решения нашей задачи можно выстроить по традиционной схеме [2], а именно множество подмножеств $SubTask_n$, где n лежит в пределах от 0 до $N-1$, разбить пополам, потом еще раз и так далее, возможно задав некоторый порог, после которого идет последовательная обработка в цикле. Этот вариант был также запрограммирован и продемонстрировал хорошее ускорение, однако его, как выяснилось, можно улучшить еще.

Модификация второго варианта распараллеливания

Причина, по которой мы еще имеем возможность увеличить эффективность параллельного исполнения нашего алгоритма, заключается в характере изменения трудоемкости перебора вариантов из множества $SubTask_n$ с увеличением n. То, что она уменьшается, очевидно, но интерес представляет характер ее зависимости от n.

Для получения ответа на этот вопрос было сгенерировано некоторое количество случайных графов с числом вершин N от 50 до 300 и вероятностью ребра p от 0.1 до 0.9. Эти же графы впоследствии использовались при изучении поведения параллельного алгоритма. На них была

собрана статистика о трудоемкости перебора $SubTask_n$. В качестве единицы измерения бралось количество рекурсивных вызовов последовательной процедуры $bron\_kerbosh()$ для выполнения работы. Типичный характер зависимости этой величины от n демонстрируется следующим графиком.

Здесь $N = 300$, верхний график соответствует значению $p = 0.5$, а нижний — $p = 0.4$. Графики наглядно демонстрируют экспоненциальный характер уменьшения трудоемкости подзадач $SubTask_n$ с увеличением n. Для больших значений p скорость уменьшения трудоемкости еще выше. По этой причине разбивать работу в соответствии со стандартной схемой неразумно, поскольку емкость листьевых активаций сильно разнится и возникает необходимость многочисленных передач кусков вычислений с модуля на модуль.

Предлагаемая модификация алгоритма состоит в том, что работа сначала делится на множества $SubTask_n$ с четным и нечетным значением n, на следующем уровне в одну группу попадают уже множества $SubTask_n$ с одинаковым остатком от деления n на 4 и так далее. После того как количество порожденных активаций достигнет заданного программистом уровня, начинается обычный цикл. Такое деление приводит к достаточно равномерному распределению работы по листьевым активациям; результаты численного эксперимента, приводимые ниже, относятся именно к этому варианту алгоритма.

Обсуждение результатов эксперимента

В процессе тестирования использовались компьютеры на базе четырехядерного процессора Intel Core i3 с тактовой частотой 3.07 GHz и 4 GB оперативной памяти, работающие под управлением 64-разрядной ОС Windows 7. Пропускная способность сети 100 Mb/s.

Описанный выше алгоритм был реализован на языке C# с использованием библиотек [4, 5]. В следующей таблице приводятся данные

Граф на 300 вершинах, вероятность ребра $p = 0.6$, Время работы последовательного алгоритма (в мс) 258478			
Кол-во ПМ	Время в мс	Ускор. (отн. парал.)	Ускор. (отн. посл.)
1	335899	1	0.769511
2	82286	4.082092	3.141215
4	49402	6.7993	5.232136
8	36838	9.118275	7.016613
12	30522	11.00514	8.46858
Граф на 300 вершинах, вероятность ребра $p = 0.7$, Время работы последовательного алгоритма (в мс) 10620270			
Кол-во ПМ	Время в мс	Ускор. (отн. парал.)	Ускор. (отн. посл.)
1	12192987	1	0.871015
2	3245722	3.756633	3.272082
4	1659134	7.349007	6.401092
8	1131466	10.77627	9.386292
12	900283	13.5435	11.79659

о времени выполнения алгоритма в зависимости от количества задействованных рабочих станций. Надо иметь в виду, что при повторении эксперимента разброс полученных временных показателей в пределах 10 % является вполне обычным.

Можно отметить, что в ряде случаев ускорение по отношению ко времени работы не только параллельного алгоритма на одном компьютере, но и последовательного его варианта превышает количество задействованных ПМ. Это объясняется в основном тем, что параллельный алгоритм, в отличие от последовательного, использует многоядерность компьютера. Кроме того, свою лепту в ускорение вносит и тот факт, что информация о текущем рекорде передается немедленно всем исполнителям и значительное количество неперспективных ветвей вычислений отсекается раньше. Поэтому даже при запуске на одном компьютере двух экземпляров приложения они выполняют работу быстрее, чем один экземпляр.

Ссылки

1. *Bron C., Kerbosh J.* Algorithm 457 – Finding all cliques of an undirected graph // Comm. of ACM. 1973. № 16. P. 575–577.

2. *Васильчиков В. В.* Средства параллельного программирования для вычислительных систем с динамической балансировкой загрузки. Ярославль : ЯрГУ, 2001.

3. *Бойцов Е. А., Васильчиков В. В.* Решение задачи о клике на языке rpC с помощью библиотеки RPM // Заметки по информатике и математике : сб. ст. Ярославль, 2011. С. 28–37.

4. *Васильчиков В. В.* Коммуникационный модуль для организации полносвязного соединения компьютеров в локальной сети с использованием .NET Framework : Свидетельство о государственной регистрации программы для ЭВМ № 2013619925, 2013.

5. *Васильчиков В. В.* Библиотека поддержки рекурсивно-параллельного программирования для .NET Framework : Свидетельство о государственной регистрации программы для ЭВМ № 2013619926, 2013.

УДК 004.65

О. В. Власова
Ярославский государственный университет им. П. Г. Демидова
E-mail: vlasova_ov@mail.ru

Типичные ошибки, возникающие при выполнении индивидуального задания по проектированию базы данных

Проектирование базы данных — процесс, требующий значительного внимания со стороны студента, так как ему необходимо учесть все нюансы дальнейшей работы с базой данных. Ошибки, допущенные на этом этапе, могут вызвать сложность и даже невозможность сопровождения базы данных в дальнейшем.

Библиография: 3 названия.

Ключевые слова: базы данных, нормализация отношений, первичный ключ, внешний ключ.

На сегодняшний день в информационных системах по ряду важнейших причин доминирующее положение занимают реляционные базы данных. Но далеко не каждая база данных обладает такими характеристиками, как отсутствие избыточности и логичность выбора структуры. Поэтому важнейшее место в жизненном цикле базы данных занимает ее проектирование. Структурная часть реляционной базы данных — это конечный (ограниченный) набор взаимосвязанных отношений (специальным образом организованных двумерных таблиц). Отношение используется для представления объектов, а также для представления связей между объектами.

Процесс проектирования БД можно разбить на несколько этапов (см. [1, 2]):

- исследование предметной области,
- создание инфологической модели,
- построение реляционной модели,

- проверку требований нормальных форм,
- описание реляционной модели на языке SQL.

К сожалению, исследование предметной области (ПО) с целью выявления всех объектов системы, логики их взаимодействия, потоков передаваемой информации не может быть проведено студентом в полном объёме в силу ряда объёктивных причин. И зачастую описание ПО дается студенту в упрощенной форме в виде готового задания.

Инфологическая модель создается по результатам проведения исследований предметной области или по готовому заданию. Инфологическая модель представляет собой описание будущей базы данных, представленное с помощью естественного языка, формул, графиков, диаграмм, таблиц и других средств, понятных как разработчикам БД, так и обычным пользователям. В настоящее время одним из наиболее широко распространенных подходов, применяемых при инфологическом моделировании, является подход, основанный на применении диаграмм «сущность-связь» (ER — Entity Relationship). Основными ошибками на этом этапе является неправильный выбор сущностей, идентифицирующих атрибутов (естественный ключ, Primary Key), их неоправданная замена суррогатными ключами, а также неверное установление связей между сущностями.

Например, при описании некоторой ПО студент выделяет сущность СОТРУДНИК со следующими атрибутами: серия и номер паспорта, фамилию, имя, отчество владельца паспорта, место и дату выдачи документа и т. п. Совокупность серии и номера паспорта однозначно определяет экземпляр данной сущности. Но выбор данных атрибутов в качестве первичного ключа может привести к ошибкам в процессе эксплуатации. Что произойдёт, если служащий сменит паспорт или предъявит иной, допускаемый в данной ситуации законом документ? Предположим, что ранее в платёжных документах отмечалась информация, соответствующая атрибутам старого паспорта. Если мы сменим значения атрибутов (серия и номер паспорта) на новые во всех сущностях (таблицах БД), то очевидно, что документы, выданные до смены паспорта, будут искажены. Искажения связаны с тем, что информация в документах, хранимых в БД, изменится, следуя требованиям ссылочной целостности. Однако информация в реальных документах сохранится в первоначальном виде. Теперь информация в БД не будет соответствовать реальной информации. Не изменять информацию в связанных сущностях тоже невозможно, поскольку в этом случае произойдёт нарушение ссылочной целостности. Ранее выданные документы будут ссылаться на служащего, которого уже не существует в базе данных. Получается замкнутый круг: нельзя изменить атрибуты, так как они ранее были использованы в платёжных документах, а эти документы

изменять нельзя, но, тем не менее, изменить атрибуты всё же надо, ибо реальная смена паспорта произошла. В чём состоит ошибка? В том, что паспорт был отождествлён с его владельцем, а, как позже выяснилось, у одного служащего может быть более одного паспорта. Таким образом, на этапе проектирования не было учтено, что служащие и паспорта представляют собой две разные сущности, связанные отношением «один-ко-многим».

Очень часто студенты вместо естественного первичного ключа, содержащего информацию о ПО и, следовательно, имеющего большой объём, используют суррогатный ключ (автоматически генерируемый), не содержащий в себе никакой информации. Свои действия они мотивируют тем, что экономят память и упрощают работу, связанную с возможной модификацией ключа. Но решение, основанное на введении суррогатного ключа, порождает транзитивную функциональную зависимость:

суррогатный ключ $\rightarrow$ естественный ключ $\rightarrow$ неключевой атрибут

Наличие транзитивной зависимости нарушает требования третьей нормальной формы (3НФ). Таким образом, бездумное использование суррогатных ключей, приводит к нарушению нормализации, а следовательно, и к аномалиям. Использование суррогатных ключей вместо естественных ключей усложняет БД и запросы к ней. Усложнение БД можно рассматривать с двух позиций. Во-первых, усложняются структуры БД, во-вторых, усложняется логика поддержания достоверности информации, хранимой в базе данных. Чтобы не нарушать 3НФ и ограничения предметной области, необходимо поддерживать уникальность и естественного, и суррогатного ключей одновременно. Соответственно, количество структур, с помощью которых поддерживается уникальность (как правило, уникальных индексов), увеличивается. Естественный ключ, являясь внешним ключом, содержит полезную информацию, и эта информация может быть использована в рамках ссылочного отношения, содержащего данный внешний ключ. При использовании естественного внешнего ключа можно на поле внешнего ключа наложить дополнительное ограничение диапазона значений. Использование естественного внешнего ключа в ряде случаев позволяет избежать соединений нескольких таблиц при выполнении запросов. Применение естественных ключей оправдывает себя и в том случае, если необходимо синхронизировать информацию в нескольких БД. Поскольку естественные ключи являются составной частью предметной области, то они сохраняют своё значение безотносительно к тому, в каком количестве БД (или одной распределённой базе данных) реализована ПО. В отличие от естественных ключей, суррогатные ключи вырабатываются каждой БД (или каждым узлом распределённой БД) самостоятельно. Поэтому при переносе информации из одной базы данных в другую, «старые» сур-

рогатные ключи, взятые из исходной БД, могут конфликтовать с суррогатными ключами в приёмной БД.

На основании изложенного выше можно сделать вывод, что использование суррогатных ключей оправдано в двух случаях:

- отсутствие естественного ключа,
- ограничение на размер первичного ключа, накладываемое в конкретной СУБД.

При переводе инфологической модели в датологическую студенты часто забывают правила формирования связей с помощью внешних ключей (Foreigin Key):

- если сущности связаны обязательной связью, то все ключевые атрибуты родительской сущности мигрируют в состав первичного ключа дочерней сущности,
- если сущности связаны необязательной связью, то все ключевые атрибуты родительской сущности мигрируют в состав неключевых атрибутов дочерней сущности.

И зачастую их отношения вообще никак не связаны.

Отношения, полученные при построении датологической модели, должны быть обязательно проверены на соответствие, как минимум, 3НФ. Однако в действительности часто оказывается, что не всегда даже условия 1НФ выполняются правильно. Яркий пример нарушения — отношение Клиент (Номер, ФИО, адрес, платеж1, платеж2, ..., платеж12). Всегда ли имеется 12 платежей? Существенен ли их порядок? Имеет ли NULL - значение смысл «неизвестно» (пока еще не заполнен), или это означает пропущенную оплату? И когда оплата была сделана?

Оплата не является характеристикой Клиента и не должна храниться в этой таблице.

Детали платежей должны храниться в отдельной таблице, в которую можно также записывать дополнительную информацию об оплате, например, когда оплата была сделана:

Клиент (Номер, фио, адрес)

Платеж ([№ платежа], [Номер клиента], дата, платеж).

Ещё одной распространенной ошибкой является неиспользование студентами средств SQL для поддержания целостности данных при описании реляционной базы данных на языке SQL. Основные правила допустимости NULL - значений, длины строки, назначения внешних ключей, правила модификации первичных ключей и так далее, все должны быть определены в базе данных. Есть много различных способов импортировать данные в СУБД. Если основные правила определены в

самой БД, то можно гарантировать, что они никогда не будут обойдены, и, следовательно, можно писать запросы, совершенно не беспокоясь о корректности данных.

Данные примеры с типичными ошибками наглядно показывают, насколько важно тщательное выполнение всех этапов проектирования БД. Правильный проект уменьшает количество ошибок, возникающих при эксплуатации базы данных.

При подготовке работы использованы материалы сайта фирмы Alexus Software[1].

Ссылки

1. *Дейт К. Дж.* Введение в системы баз данных. 8-е изд./ пер. с англ. М. : Вильямс, 2005.

2. *Власова О. В.* Системы управления базами данных : лабораторный практикум. Ярославль: ЯрГУ, 2010.

[1] http://www.alexus.ru/russian/articles/dbms/keys/index.htm

УДК 517.9

С. Д. Глызин, А. Ю. Колесов
Ярославский государственный университет им. П. Г. Демидова
E-mail: c_d_glyzin@mail.ru
E-mail: kolesov@uniyar.ac.ru

Релаксационные модели динамики нейронных ассоциаций

Представлен учебный курс, посвященный теории релаксационных колебаний для специального класса уравнений с запаздываниями, моделирующих электрическую активность нервных клеток. Систематически изложен новый способ описания феномена «bursting behavior» и феномена буферности в нейронных системах, использующих уравнения с запаздыванием. Рассматриваются сингулярно возмущенные скалярные нелинейные дифференциальные уравнения с двумя запаздываниями и их системы, являющиеся математической моделью нейронных ассоциаций. Установлено, что в «длинной» одномерной цепочке диффузионно связанных нейронов такого типа при согласованном увеличении числа звеньев цепочки и уменьшении коэффициента диффузии происходит неограниченный рост количества сосуществующих устойчивых периодических движений с любым наперед заданным количеством «интенсивных» всплесков на периоде.

Библиография: 4 названия.

Ключевые слова: нейродинамика, дифференциально-разностные уравнения, релаксационные колебания, устойчивость.

Моделирование динамики изменения электрического потенциала нервных клеток связано, в первую очередь, с работами А. Л. Ходжкина и А. Ф. Хаксли. Этим авторам в статье [1] впервые удалось представить феноменологическую модель, полученную на основе соотношений балансного типа, так, что ее динамика при надлежащем выборе параметров обладает основными качественными свойствами, характерными для наблюдаемых в эксперименте нервных клеток. Во многих случаях данная модель имеет вполне удовлетворительное не только качественное, но и количественное соответствие экспериментальным данным.

Модель Ходжкина–Хаксли довольно сложна, и потому с момента ее появления предпринимались многочисленные попытки ее упрощения с сохранением основных эффектов, характерных для динамики нейронов. В суммирующих статьях [2, 3] приведен ряд требований, которым должна удовлетворять модель импульсного нейрона, и перечислено большое число модельных систем. Среди этих требований наиболее важным является существование у модели устойчивого периодического режима импульсного типа. Другим существенным требованием является наличие у модели (при некоторых значениях параметров) bursting-эффекта.

Для формирования математической модели одиночного импульсного нейрона воспользуемся рассуждениями, аналогичными применяемым Ходжкиным и Хаксли в [1]. Примем во внимание только калиевые и натриевые токи, в качестве начала отсчета возьмем уровень наибольшей поляризации мембраны, а отклонение потенциала от этого уровня обозначим $u(t)$. Уравнение баланса токов (если пренебречь токами утечки) записывается в этой ситуации в виде

$$c\dot{u} = I_{\mathrm{Na}} + I_{\mathrm{K}}, \tag{1}$$

где коэффициент $c > 0$ обычно называют емкостью мембраны.

Для формирования содержательной модели сделаем несколько дополнительных предположений.

Условие 1. *Считаем, что токи I_{Na} и I_K можно представить следующим образом:*

$$I_{Na} = \chi_{Na}(u) \cdot u, \qquad I_K = \chi_K(u) \cdot u, \tag{2}$$

где $\chi_{Na}(u)$ и $\chi_K(u)$ – функции, определяющие натриевую и калиевую проводимости.

Условие 2. *Считаем, что $\chi_{Na}(0) = -\alpha_0$, и при $u \to \infty$ $\chi_{Na}(u) \to -\beta_0$, причем $\beta_0 > \alpha_0$, где α_0, β_0 – положительные константы.*

Отрицательность $\chi_{\mathrm{Na}}(0)$ объясняется тем, что в состоянии сильной поляризации ($u \ll 1$) на внутренней поверхности мембраны наблюдается избыток ионов натрия. В их откачивании из клетки и состоит работа ионных насосов (см. [1]). В силу положительности заряда ионов натрия данный процесс уменьшает мембранный потенциал, а потому $\chi_{\mathrm{Na}}(u) < 0$ при $u \ll 1$.

Перейдем к обсуждению зависимости $\chi_{\mathrm{K}}(u)$. В состоянии сильной поляризации поток ионов калия направлен внутрь клетки, что способствует росту мембранного потенциала, а поэтому $\chi_{\mathrm{K}}(u) > 0$ при $u \ll 1$. Однако после прохождения пика потенциала поток ионов калия меняет свое направление. Следовательно, существует такой уровень потенциала, что при значениях u выше этого уровня проводимость $\chi_{\mathrm{K}}(u) < 0$. Таким образом, приходим к следующему условию.

Условие 3. *Полагаем* $\chi_K(0) = \alpha_1$, *и* $\chi_K(u) \to -\beta_1$ *при* $u \to \infty$, *где* α_1, β_1 – *также положительные константы.*

Важной характеристикой ионных каналов является запаздывание величины их проводимости по времени.

Условие 4. *Будем считать, что величина калиевой проводимости запаздывает по отношению к текущему значению мембранного потенциала, и примем это запаздывание за единицу времени, то есть полагаем, что* $\chi_K = \chi_K(u(t-1))$, *запаздывание величины натриевой проводимости примем за* $0 < h < 1$ *и положим, что* $\chi_{Na} = \chi_{Na}(u(t-h))$.

Отметим, кроме того, что в состоянии сильной поляризации мембранный потенциал должен расти, поэтому $\chi_{\mathrm{Na}}(0) + \chi_{\mathrm{K}}(0) > 0$.

При условиях 1–4 из (1) получаем

$$c\dot{u} = [\chi_{\mathrm{Na}}(u(t-h)) + \chi_{\mathrm{K}}(u(t-1))]u. \tag{3}$$

Нетрудно видеть, что модель (3) при $h = 1$ сводится к обобщенному уравнению Хатчинсона. Отметим, что аналогичным образом получена модель с одним запаздыванием в книге [4].

Для формирования модельного уравнения обозначим

$$\begin{aligned} &\chi_{\mathrm{Na}}(u) = (\chi_{\mathrm{K}}(0) + \chi_{\mathrm{Na}}(0))f(u) - \chi_{\mathrm{K}}(0), \\ &\chi_{\mathrm{K}}(u) = \chi_{\mathrm{K}}(0) - (\chi_{\mathrm{K}}(0) + \chi_{\mathrm{Na}}(0))g(u), \quad \lambda = (\chi_{\mathrm{K}}(0) + \chi_{\mathrm{Na}}(0))/c. \end{aligned} \tag{4}$$

Тогда от (3) приходим к уравнению

$$\dot{u} = \lambda[f(u(t-h)) - g(u(t-1))]u. \tag{5}$$

Здесь $u(t) > 0$ – мембранный потенциал нейрона, параметр $\lambda > 0$ характеризует скорость протекания электрических процессов в системе и предполагается большим, а параметр h фиксирован и принадлежит интервалу $(0, 1)$. Относительно функций $f(u)$, $g(u) \in C^1(\mathbb{R}_+)$, $\mathbb{R}_+ = \{u \in \mathbb{R} : u \geq 0\}$, предполагаем, что они обладают свойствами:

$$\begin{aligned} &f(0) = 1, \ \ g(0) = 0; \ \ f(u) = -a_0 + O(1/u), \ \ g(u) = b_0 + O(1/u), \\ &uf'(u) = O(1/u), \ \ ug'(u) = O(1/u), \\ &u^2 f''(u) = O(1/u), \ \ u^2 g''(u) = O(1/u) \ \text{ при } \ u \to +\infty, \end{aligned} \tag{6}$$

где $a_0 = -\dfrac{\alpha_1 - \beta_0}{\alpha_1 - \alpha_0}$, $b_0 = \dfrac{\alpha_1 + \beta_1}{\alpha_1 - \alpha_0}$ – положительные константы.

Основные результаты, обсуждаемые в представленном курсе, касаются релаксационных свойств уравнений (5), а также систем связанных уравнений такого типа. Важно отметить, что полученная модель является вполне содержательной, поскольку при подходящем выборе параметров она обладает как режимами с одним всплеском на периоде

(например, при $h = 1$), так и любым наперед заданным количеством таких всплесков. В частности, показано, что по любому фиксированному натуральному n можно так подобрать фигурирующие в (5), (6) параметры h, a_0, b_0, что при всех достаточно больших λ уравнение (5) будет иметь экспоненциально орбитально устойчивый цикл $u = u_*(t, \lambda)$ периода $T_*(\lambda)$, где $T_*(\lambda)$ при $\lambda \to \infty$ стремится к некоторому конечному пределу $T_* > 0$. При этом функция $u_*(t, \lambda)$ на отрезке времени длины $T_*(\lambda)$ допускает ровно n подряд идущих асимптотически высоких всплесков, а все остальное время – асимптотически мала, тем самым при указанном выборе параметров $u_*(t, \lambda)$ является bursting-циклом.

Рассмотрим теперь цепочку из m, $m \geq 2$ нейронов вида (5), каждый из которых взаимодействует с двумя ближайшими своими соседями. В этом случае вместо (5) получается система

$$\dot{u}_j = d\,(u_{j+1}-2u_j+u_{j-1})+\lambda[f(u_j(t-h))-g(u_j(t-1))]u_j,\ \ j = 1, \ldots, m, \quad (7)$$

где $u_0 = u_1$, $u_{m+1} = u_m$, а параметр $d > 0$ порядка единицы характеризует глубину связи между нейронами. Основные результаты обсуждаемые в предлагаемом курсе, касаются релаксационных свойств уравнений (5) и систем связанных уравнений (7).

Опишем общую структуру курса. В первой части формулируется общий подход к асимптотическому интегрированию уравнения (5) с одним запаздыванием и обосновывается существование и устойчивость его релаксационных циклов. Затем рассматривается цепочка связанных осцилляторов такого типа. Связь в этом случае предполагается диффузионной. При подходящих значениях параметров в цепочке из m элементов удается доказать сосуществование не менее m релаксационных циклов. Вторая часть курса посвящена анализу уравнения (5) с двумя запаздываниями, в ней излагается механизм возникновения bursting-цикла. Наряду с этим удается показать, что в системе из m связанных генераторов при подходящем выборе параметров сосуществует не менее m устойчивых bursting-циклов. В третьей части курса обсуждается модель синаптической связи между нейронами. Наконец, в последней части методы, разработанные для моделей (5), (7), применяются к сетям Хопфилда с запаздыванием. Для таких сетей получены результаты, аналогичные найденным для моделей (5), (7).

Ссылки

1. *Hodgkin A. L., Huxley A. F.* A quantitative description of membrane current and its application to conduction and excitation in nerve // J. Physiol. 1952. V. 117. P. 500–544.

2. *Izhikevich E.* Neural excitability, spiking and bursting // International Journal of Bifurcation and Chaos. 2000. V. 10(6). P. 1171–1266.

3. Dynamical principles in neuroscience / M. I. Rabinovich et al. // Rev. Mod. Phys. 2006. V. 78. P. 1213–1265.

4. *Кащенко С. А., Майоров В. В.* Модели волновой памяти. М. : Эдиториал УРСС, 2009. 288 с.

УДК 371.214, 371.321

О. А. Дунаева
Ярославский государственный университет им. П. Г. Демидова
E-mail: OlyaDy@gmail.com

Принципы отбора тем для курса «Алгоритмические основы мультимедийных технологий»

Обсуждаются вопросы формирования рабочей программы курса «Алгоритмические основы мультимедийных технологий», который читается для магистрантов второго курса. Дается обзор актуальных алгоритмических проблем в области мультимедийных технологий. Описывается подход автора к выбору тем, которые легли в основу курса, читаемого магистрантам на факультете ИВТ. Проводится сравнение с другими вариантами учебной программы данного курса.

Ключевые слова: мультимедийные технологии, семантический анализ, цифровая обработка сигналов.

В последнее десятилетие в связи с широким распространением цифровых технологий фото- и видеосъемки, а также в связи с увеличением пропускной способности каналов связи существенно увеличился поток видео- и аудиоданных, что сделало еще более актуальными задачи хранения, передачи и обработки мультимедийных данных.

В целом задачи обработки мультимедийных данных можно разделить на три больших класса: 1) задачи, связанные с хранением и передачей мультимедийных данных; 2) задачи, связанные с преобразованием мультимедийных данных; 3) задачи, связанные с выделением из мультимедийных данных семантической информации.

Задачи хранения и передачи мультимедийных данных решаются за счет сжатия — уменьшения объема данных без потери или с потерей качества. Для задачи сжатия мультимедийных данных существует ряд известных алгоритмов, которые (особенно в случае сжатия с потерей качества) учитывают особенности восприятия информации человеком. Так, на изображении человек различает различные уровни яркости гораздо лучше, чем различные цвета, поэтому при сжатии с потерями

выгоднее жертвовать цветовой информацией, чем информацией о яркости. При сжатии с потерями аудиоданных применяется психоакустическая модель слуха, т. е. учитывается тот факт, что человеческое ухо не способно воспринимать частоты свыше 20 КГц и из сигнала можно исключить несущественную часть спектра таким образом, чтобы результат, с точки зрения восприятия человеком, был практически неотличим от оригинала. В последнее время большое распространение получили видеоконференции, при которых передача данных происходит в режиме реального времени, что добавляет новые требования к алгоритмам сжатия мультимедийных данных. Например, при понижении пропускной способности канала передачи данных адаптивный алгоритм сжатия может за счет снижения качества сигнала уменьшить объем потока данных с целью обеспечения корректной работы в режиме реального времени.

Преобразование мультимедийных данных чаще всего связано с цифровой обработкой сигналов, представляющих эти данные. Здесь встают такие задачи, как цветовая коррекция, улучшение контрастности, выравнивание освещенности для видеопотока и изображений; шумоподавление, передискретизация, изменения громкости, скорости проигрывания и высоты тона для аудиопотока; устранение мозаичности и муаровых эффектов для видеопотока.

Наиболее сложные задачи обработки мультимедийных данных связаны с выделением из них семантической информации, т. е. с преобразованием информации, представленной в виде звука, изображений или видеопотока в форму, пригодную для дальнейшей обработки и поиска. Так, поиск информации, который легко реализуется для текстовой информации, для фотографий, аудиоданных или тем более видеоинформации оказывается намного более сложной задачей. Сейчас ведутся активные разработки в направлении компьютерного зрения, распознавания аудиоинформации, классификации изображений. Ищутся пути решения задач информационного поиска: для изображений и видеопотока решаются задачи распознавания лиц (с определением пола и возрастной группы), жестов и мимики лица, номерных знаков машин, поиска изображений по контексту; для аудиоданных решаются задачи распознавания музыкальных композиций и речи, задача идентификации человека по голосу. Также активно развиваются системы слежения за движущимися объектами в видеопотоке, актуальные, например, для охранных и спортивных мультимедийных систем. Так, для охранных систем представляет интерес задача автоматического выделения вещей, которые несет движущийся человек, и задача автоматического детектирования оставленных сумок. Для спортивных систем актуальны задачи автоматического слежения за мячом, вычисления статистики перемещения игроков и детектирования пересечения мячом линии ворот. Очень

активно развиваются алгоритмы автоматического распознавания речи, системы голосового ввода информации и голосового управления. Таким образом, область современных мультимедийных технологий, связанных с выделением и анализом семантической информации очень широка.

На курс «Алгоритмические основы мультимедийных технологий» отводится 72 часа, большая часть которых отведена самостоятельной работе студентов, поэтому вопрос выбора тем для аудиторных занятий является очень важным.

Большинство известных автору программ курса «Алгоритмические основы мультимедийных технологий» посвящены изучению различных алгоритмов сжатия, стандартов кодирования изображений, аудио- и видеоданных. Такие курсы отличаются лишь расстановкой акцентов. В некоторых курсах глубже рассматриваются математические аспекты работы алгоритмов сжатия, в других же акцент делается на их реализации. В некоторых вариантах курса дополнительно рассматриваются общие вопросы цифровой обработки сигналов (спектральные методы, задачи шумоподавления, изменения частоты дискретизации сигнала, алгоритмы обработки речевых сигналов и т. д.), в других случаях дополнительно рассказывается о 3D графике (построение трёхмерных сцен с использованием OpenGL, понятия текстуры и шейдера).

В нашем курсе после изложения необходимого вводного материала основной акцент делается на алгоритмах семантического анализа мультимедийных данных [1], что позволяет магистрантам прикоснуться к одной из наиболее бурно развивающихся областей мультимедийных технологий. Приведем примерный список задач, которые рассматриваются в курсе, читаемом автором для магистрантов специальностей 010300.68 «Фундаментальная информатика и информационные технологии» и 010400.68 «Прикладная математика и информатика»:

- Алгоритмы выделения границ на изображении. Понятие градиента, методы его приближенного вычисления. Детектор краев Canny.

- Понятие оптического потока. Задача вычисления оптического потока по видеопотоку. Проблема апертуры, варианты выбора особых точек на кадре.

- Системы видеонаблюдения и задачи автоматизации видеомониторинга. Методы моделирования неподвижного фона. Понятие маски переднего плана.

- Детектор объектов Viola-Jones, основные идеи алгоритма. Система признаков Хаара. Интегральное изображение, алгоритм построения интегрального изображения. Идея алгоритма AdaBoost.

- Особенности звуковосприятия человека. Диапазон частот, доступных человеческому слуху. Временны́е характеристики слухового восприятия. Эффект маскировки, бинауральный эффект.

- Спектрограмма звукового сигнала. Пики спектрограммы, алгоритмы формирования аудиоотпечатка. Алгоритм Shazam распознавания музыкальных композиций в аудиопотоке.

Ясно, что полная рабочая программа курса включает в себя также вводные вопросы, необходимые для понимания перечисленных выше тем, но в целом основное внимание в курсе уделяется задачам информационного поиска и семантического анализа мультимедийных данных. На наш взгляд, именно эти вопросы являются наиболее актуальными и именно их рассмотрение позволяет магистрантам получить адекватное представление о современном состоянии и развитии мультимедийных технологий.

Ссылки

1. *Шапиро Л., Стокман Дж.* Компьютерное зрение. М. : Бином, 2006. 752 с.

УДК 510.21

В. Г. Дурнев
Ярославский государственный университет им. П. Г. Демидова
E-mail: durnev@univ.uniyar.ac.ru

Узлы и косы – простые числа и гипотеза Римана – комбинаторная теория групп и криптография

В заметке рассматриваются некоторые связи между такими, казалось бы, весьма далекими друг от друга математическими дисциплинами учебных планов, как «Топология», «Теория чисел», «Комплексный анализ», «Комбинаторная теория групп» и «Криптография».

Библиография: 20 названий.

Ключевые слова: простые числа, гипотеза Римана, диофантовы уравнения, комбинаторная теория групп.

Килиманджаро – покрытый вечными снегами горный массив высотой в 19710 футов, как говорят, высшая точка Африки. Племя масаи называет его западный пик «Нгайэ-Нгайя», что значит «Дом бога». Почти у самой вершины западного пика лежит иссохший мерзлый труп леопарда. Что понадобилось леопарду на такой высоте, никто объяснить не может.

Э. Хемингуэй. «Снега Килиманджаро».

«Фактор Мэлори». Отвечая на вопрос «Нью-Йорк таймс», почему ему так хочется забраться на Эверест, Джордж Мэлори ответил «Потому, что она есть».

Обучение редко приносит плоды кому-либо, кроме тех, кто предрасположен к нему, но им оно почти не нужно.

Гиббонс

Заметка ориентирована на студентов специальности «Компьютерная безопасность», учебный план по которой не включает дисциплину «Топология», поэтому не предполагается знакомство читателей с такими понятиями, как топологическое пространство и его фундаментальная группа, узлы и их группы, задание групп узлов образующими элементами и определяющими соотношениями. Это потребовало привести вначале соответствующие определения, но не в самой общей ситуации, а адаптированно к целям заметки.

Узлы и косы

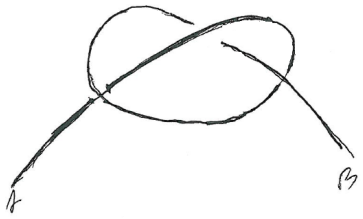

Рис. 1.

Рис. 2.

Многие из нас умеют завязывать *узлы* на неслишком короткой веревке, но не все – у некоторых, в том числе и у автора, шнурки на ботинках постоянно развязываются. Простейший узел изображен на рис. 1. Интуитивно ясно, что если закрепить концы веревки A и B, то этот узел, не разорвав веревку, нельзя развязать, т. е. преобразовать в *тривиальный* узел, изображенный на рис. 2.

Вместо закрепления концов веревки A и B их можно просто склеить, и тогда узел с рис. 1 превратится в узел на рис. 3, называемый *клеверным листом* или *трилистником*, а «незаузленный» узел с рис. 2 превратится в окружность с рис. 4.

Если обычную окружность с рис. 4 назвать «незаузленной» окружностью, то узел с рис. 3 можно назвать «заузленой окружностью»: если на узлы с рис. 3 и рис. 4 посмотреть как на ходы, которые проел червяк

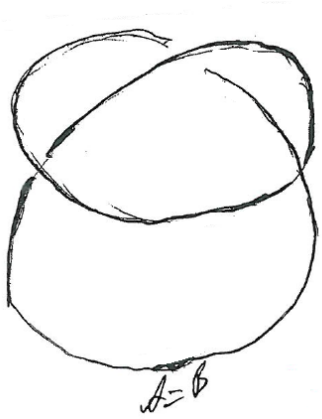

Рис. 3.

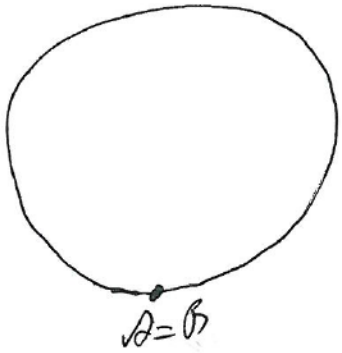

Рис. 4.

в яблоке, то «червяк их не сможет различить», т. е. «сами по себе» и узел с рис. 3 и узел с рис. 4 – это просто окружности, но они по разному расположены в пространстве R^3. Чтобы сделать узел объектом математического изучения требуется ввести адекватную математическую замену рассмотренных «физических» объектов. Узел в R^3 – это замкнутая кривая без самопересечений.

Мы приходим к следующему определению узла в R^3:

узел в R^3 – это образ $f(S)$ – единичной окружности S относительно инъективного, непрерывного отображения f этой окружности в R^3.

Однако это слишком общее определение, как и определение непрерывной кривой, под которое подходит кривая Пеано, заполняющая весь единичный квадрат. Поэтому понятие непрерывной кривой обычно сужается до понятия гладкой кривой. Так и в случае узлов обычно ограничиваются изучением так называемых полигональных узлов – узлов, образованных замкнутыми ломаными линиями в R^3 без самопересечений.

Математическая теория узлов начала интенсивно развиваться начиная с работы Листинга 1848 года [1]. Большой вклад в теорию узлов в первой половине XX века внесли Виртингер, Дэн, Александер, Рейдемейстер и Зейферт. Узлы относятся к так называемой «топологии малой размерности».

Может быть, самый известный узел – это Гордиев узел, с которым связана легенда и известное выражение «Разрубить гордиев узел». Легенда гласит, что фригийский царь Гордий завязал весьма сложный узел, а жрецы Фригийского храма Зевса предсказали, что первый, кто развяжет этот узел, будет самым выдающимся царем, ему покорится весь мир, он создаст империю, охватывающую всю Азию. По мнению некоторых авторов, легенда гласит (в изложении легенды некоторыми авторами), что покоривший столицу Фригии великий полководец древности Александр Македонский, войдя в древний храм, без долгих размышлений выхватил мечь и рассек одним ударом гордиев узел (таким виделся излагавшим этот вариант древней легенды идеал правителя?). Это решительное, но необдуманное действие истолковали жрецы: «Он завоюет мир! Но мечом, а не дипломатией». Однако другие авторы утверждают (в изложениии легенды другими авторми), что Александр Макенонский не разрубил узел мечом, а решил задачу (проблему) – вынул закреплявший яремный ремень крюк – «гестор» и узел развязался! (таким виделся излагавшим в таком варианте легенду идеал правителя?) На этом историческом примере можно было бы порассуждать о взаимотношениях в истории «метода грубой силы» и «интеллектуального метода» – «прыгать» или «думать», что можно было бы увязать и с вопросом обеспечения информационной безопасности, но, может быть, сделаем это в другое время и в другом месте.

Рассмотрим еще один достаточно интересный объект «топологии малой размерности» – *косу*. В наш «век коротких стрижек» может быть многие и забыли о косах, некогда украшавших женские головы. Пример косы на двух нитях приведен на рис. 5 и рис. 6.

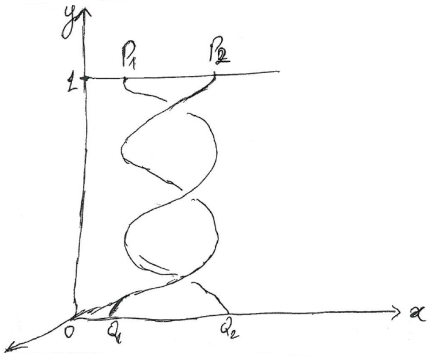

Рис. 5.

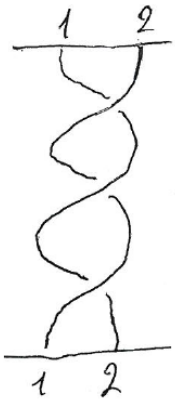

Рис. 6.

Этот пример косы не случайный – коса с рис. 5 и рис. 6 тесно связана с узлом трилистником с рис. 3, но об этом позже.

Как и в случае узлов, для математического изучения кос нам потребуется заменить «реальные» косы на математичские косы, которые мы будем по-прежнему называть просто *косами*.

Для построения *кос* на n нитях фиксируем в пространстве R^3 два набора из n точек каждый:

$$P_1 = (1,0,1), P_2 = (2,0,1), \ldots, P_n = (n,0,1) \quad \text{и}$$
$$Q_1 = (1,0,0), Q_2 = (2,0,0), \ldots, Q_n = (n,0,0).$$

Тогда коса – это набор из n непрерывных инъективных отображений $f_1(t)$, $f_2(t)$, ..., $f_n(t)$ единичного отрезка $[0,1]$ в пространство R^3 таких, что при любом i $(1 \le i \le n)$:

$f_i(0) = P_i$ и существует такое j_i, что $f_i(1) = Q_{j_i}$ причем числа j_1, ..., j_n образуют перестановку чисел 1, ..., n, кроме того,

если $f_i(t) = (f_i^{(1)}(t), f_i^{(2)}(t), f_i^{(3)}(t))$, то $f_i^{(3)}(t))$ – монотоннно убывающая функция,

при $i \ne j$ $f_i([0,1]) \cap f_j([0,1]) = \emptyset$.

То есть функция $f_i(t)$– это «непрерывная нить, идущая монотоннно сверху вниз из точки P_i в точку Q_{j_i}, а последнее условие означает, что нити попарно не пересекаются». Второе условие гарантирует монотонное спускание нити вниз.

На самом деле коса определяется с точностью до некоторой эквивалентности: две косы B и B_1 на n нитях называются эквивалентными (не различаются, считаются одной и той же косой), если существует гомеоморфизм f пространства R^3, т. е. такое биективное отображение этого пространства на себя, что f и f^{-1} непрерывны, ограничения f на подпространства $\{(x,y,z)|x \ge 1\}$ и $\{(x,y,z)|x \le -1\}$ оставляет на месте точки этих подпространств (индуцирует тождественные отображения на этих подпространствах) и $f(B) = B_1$. Но нам не потребуется это

математическое уточнее этого интуитивно ясного понятия «нити косы можно непрерывно и без склейки деформировать».

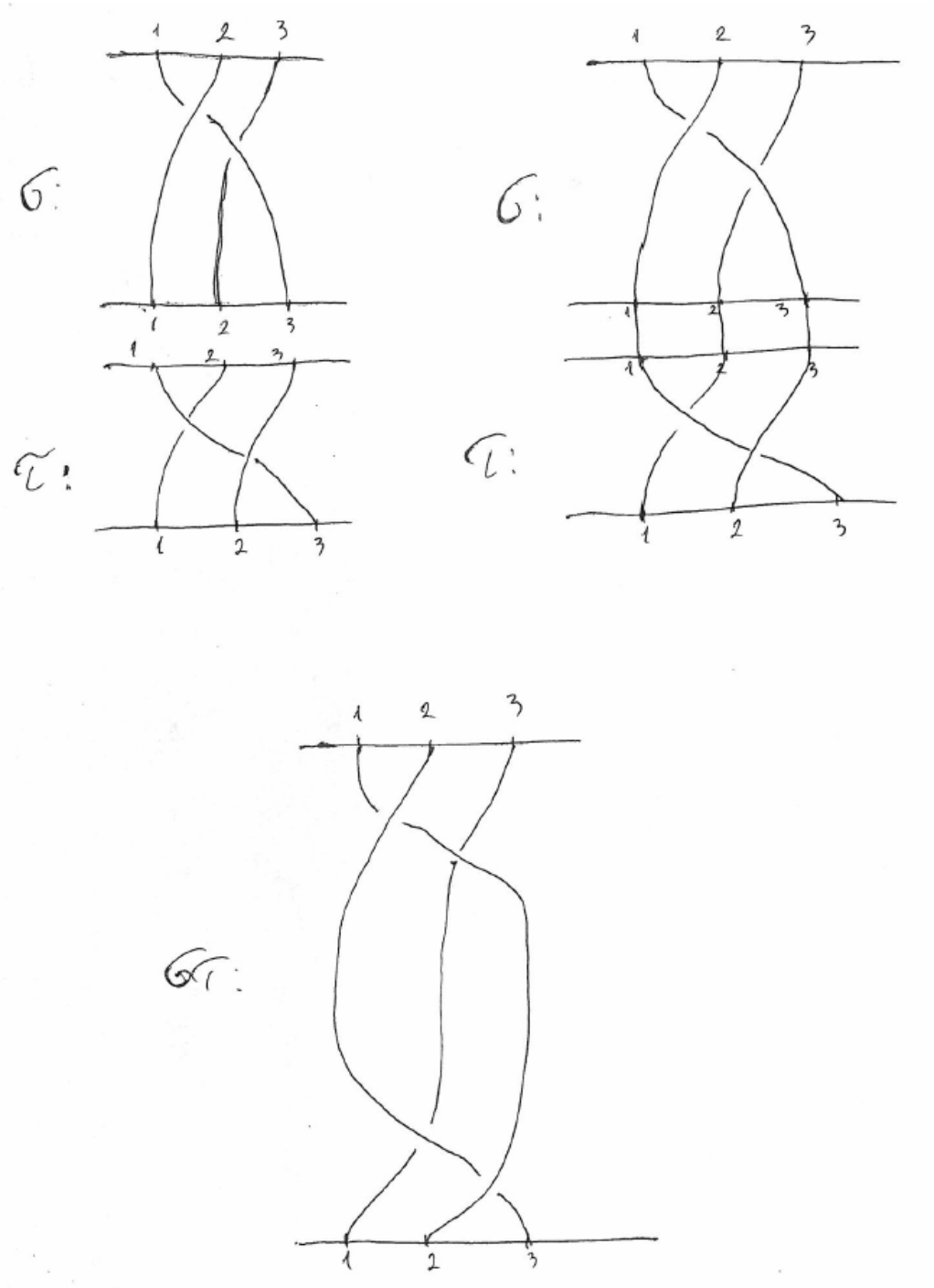

Рис. 7.

Более интересным является то, что косы можно умножать. Как это происходит, легко понять из рис. 4: чтобы косу σ умножить на косу τ,

надо просто к σ «приклеить, прикрепить» косу τ. Аналогично на множестве всех кос на n нитях можно достаточно естественным образом ввести операцию умножения кос («склейку» кос), относительно которой получим весьма интересную и важную группу $B(n)$, которая изучается уже более 60 лет – с середины XX века, является источником все новых задач и обобщений, а в начале XXI века нашла применения в криптографии [2]. Для узлов тоже можно определить бинарную алгебраическую операцию, состоящую в «привязываниии» одного узла к другому, но относительно нее группы не получим, а получим лишь коммутативную полугруппу, если примем, что существует «множество всех узлов в пространстве R^3». Но и в таком случае придется рассматривать в качестве элементов полуруппы не узлы, а классы эквивалентных узлов.

Если коса σ – это набор непрерывных инъективных отображений $f_1(t)$, $f_2(t)$, ..., $f_n(t)$ единичного отрезка $[0,1]$ в пространство R^3 таких, что при любом i $(1 \le i \le n)$: $f_i(0) = P_i$ и $f_i(1) = Q_{j_i}$, а коса τ – это набор непрерывных инъективных отображений $g_1(t)$, $g_2(t)$, ..., $g_n(t)$ единичного отрезка $[0,1]$ в пространство R^3 таких, что при любом i $(1 \le i \le n)$: $g_i(0) = P_i$ и $g_i(1) = Q_{t_i}$, то коса $\sigma\tau$, равная произведению этих кос σ и τ, – это набор непрерывных инъективных отображений $h_1(t)$, $h_2(t)$, ..., $h_n(t)$ единичного отрезка $[0,1]$ в пространство R^3 таких, что при любом i $(1 \le i \le n)$: $h_i(0) = P_i$ и

$$h(t) = \begin{cases} f_i(2t), & \text{если } 0 \le t \le 1/2; \\ g_{j_i}(2t-1), & \text{если } 1/2 \le t \le 1. \end{cases}$$

Относительно введенной операции умножения косы на n нитях образуют группу $B(n)$, которая может быть весьма несложно задана ***образующими элементами и определяющими соотношениями***, но об этом будет сказано ниже. Косы можно рассматривать как геометрические (топологические) объекты, но можно на них смотреть и как на алгебраические объекты. Алгебраическая теория кос ведет свое начало с работ Артина [3] и А. А. Маркова [4].

Узлы и косы – простейшие объекты изучения 3-мерной топологии или, как теперь принято говорить, топологии малой размерности. Известно, что ряд проблем на сегодняшний день решен для топологии достаточно большой размерности, однако для размерностей 3 и 4 при решении аналогичных вопросов нередко возникают немалые трудности. Например, проблема гомеоморфности отрицательно решена А. А. Марковым для 4-мерных многообразий [5]. А для 3-мерных многообразий лишь в 1994 году положительно решена алгоритмическая проблема распознавания стандартной 3-мерной сферы [6], [7] и совсем недавно доказана гипотеза Пуанкаре. С. П. Новиков доказал [8] алгоритмическую неразрешимость проблемы распознавания стандартной 5-мерной сферы, а для 4-мерной сферы вопрос остается открытым.

Фундаментальная группа

Напомним понятие *фундаментальной группы*, введенное А. Пуанкаре в большой работе 1895 года «Analysis Situs». А в столь же большой работе 1908 года Х. Титце установил [9], что фундаментальные группы некоторых многообразий, заданных клеточными комплексами, имеют конечные задания.

Пусть $U \subseteq R^3$ – подмножество пространства R^3.

Путь в U *– это любое непрерывное отображение* f *отрезка* $[0,1]$ *в* U. При этом точка $a = f(0)$ называется начальной точкой пути f, а $b = f(1)$ – его конечной точкой.

Некоторые пути можно *умножать*: если f и g – такие два пути, что начальная точка пути g совпадает с конечной точкой пути f, т. е. $g(0) = f(1)$, то ***произведением путей*** f ***и*** g называется путь $f * g$, определяеиый равенствами

$$(f * g)(t) = \begin{cases} f(2t), & \text{если } 0 \le t \le 1/2; \\ f(2t-1), & \text{если } 1/2 \le t \le 1. \end{cases}$$

Зафиксируем во множестве U точку p и будем рассматривать лишь такие пути в U, которые начинаются и заканчиваются в точке p, т. е. такие непрерывные отображения f отрезка $[0,1]$ в U, для которых $f(0) = p = f(1)$. Такие пути называются *петлями* в точке p.

На множестве всех петель в точке p введем отношение *эквивалентности*, называемое гомотопической эквивалентностью.

Петли f и g во множестве U с фиксированной точкой p называются *эквивалентными*, если существует такое непрерывное отображение $F(t,s)$ единичного квадрата $[0,1] \times [0,1]$ в U, что

$$\text{при любом } t: \ F(t,0) = f(t), \ F(t,1) = g(t),$$
$$\text{при любом } s: \ F(0,s) = p = F(1,s).$$

Для произвольной петли f в точке p через $[f]$ обозначим определяемый этой петлей класс эквивалентности, состоящий из всех петель g в этой точке p, эквивалентных петле f (напомним, что мы рассматриваем лишь пути и петли в U).

Легко проверяется, что введенное отношение эквивалентности действительно является отношением эквивалентности, т.е. рефлексивно, симметрично и транзитивно, и множество классов эквивалентности с естественным образом определенным умножением

$$[f] \cdot [g] = [f * g]$$

является группой, которая называется *фундаментальной группой* или *группой Пуанкаре* множества U с фиксированной точкой p и обозначается $\pi(U,p)$. Если множество U линейно связно, т. е. любые две его

точки можно соединить непрерывным путем в U, то для любых двух его точек p и q группы $\pi(U,p)$ и $\pi(U,q)$ изоморфны. Пусть φ – путь в U, соединяющий точку q с точкой p, т.е. $\varphi(0)=q$ и $\varphi(1)=p$. Обозначим через $\overline{\varphi}$ обратный путь, т. е. для любого t $(0\le t\le 1)$ выполнено равенство $\overline{\varphi}(t)=\varphi(1-t)$. Изоморфизм φ^* группы $\pi(U,p)$ на группу $\pi(U,q)$ можно задать равенством

$$\varphi^*([f]) \;=\; [\varphi * f * \overline{\varphi}].$$

Проверка деталей предоставляется читателю.

Поэтому в случае линейно связного U точку p можно не указывать и говорить просто о *фундаментальной группе* множества $\pi(U)$.

Любому непрерывному отображению φ множества $U\subseteq R^3$ во множество $V\subseteq R^3$ соответствует естественный гомоморфизм φ_* группы $\pi(U,p)$ в группу $\pi(V,f(p))$:

$$\varphi_*([f]) \;=\; [\varphi\circ f],$$

где $\varphi\circ f$ – путь в V, определяемый равенством $(\varphi\circ f)(t)=\varphi(f(t))$. При этом если φ – гомеоморфизм множества $U\subseteq R^3$ на множество $V\subseteq R^3$, т. е. такое биективное отображение, что оно само и ему обратное φ^{-1} непрерывны, то φ_* – изоморфизм группы $\pi(U,p)$ на группу $\pi(V,f(p))$. Поэтому *если линейно связные множества U и V гомеоморфны, то их фундаментальные группы $\pi(U)$ и $\pi(V)$ изоморфны.*

С точки зрения приложений более полезным оказывается обратное утверждение:

если фундаментальные группы $\pi(U)$ и $\pi(V)$ линейно связных множеств U и V неизоморфны, то сами множества негомеоморфны.

Узлы $K_1\subseteq R^3$ и $K_2\subseteq R^3$ называются ***эквивалентными***, *если существует гомеоморфизм f пространства R^3 такой, что $f(K_1)=K_2$.* Мы не интересуемся вопросом, сохраняет ли гомеоморфизм f ориентацию пространства R^3. Так как $f(R^3\setminus K_1)=R^3\setminus K_2$, то f_* – изоморфизм фундаментальных групп $\pi(R^3\setminus K_1)$ и $\pi(R^3\setminus K_2)$. Поэтому *если группы двух узлов неизоморфны, то сами узлы неэквивалентны.* Это открывает путь доказательства неэквивалентности узлов и делает естественным связать с узлом $K\subseteq R^3$ группу $G(K)=\pi(R^3\setminus K)$, называемую *группой узла K*.

Виртингер [10] установил, что если узел K допускает «достаточно хорошую» проекцию на плоскость, то можно найти достаточно простое задание его группы $G(K)$ ***образующими элементами и определяющими соотношениями*** и попытаться применить алгебраические методы для доказательства неэквивалентности узлов K_1 и K_2 – через доказательство неизоморфности их групп $G(K_1)$ и $G(K_2)$.

Задание групп образующими элементами и определяющими соотношениями

Виртингер опирался на введенный Вальтером фон Диком в 1882 – 1883 годах способ *задания групп образующими элементами и определяющими соотношениями*. Такой способ задания групп возникает естественным образом в топологии как способ задания фундаментальных групп некоторых топологических пространств.

Пусть $\mathcal{A} = \{a_1, \ldots, a_n\}$ – произвольный конечный алфавит. Введем алфавит букв-двойников $\mathcal{A}^{-1} = \{a_1^{-1}, \ldots, a_n^{-1}\}$. Объединение $\mathcal{A} \cup \mathcal{A}^{-1}$ этих алфавитов будем называть *групповым алфавитом*.

Зафиксируем некоторый конечный набор упорядоченных пар слов $\langle A_1, B_1\rangle$, $\ldots$, $\langle A_m, B_m\rangle$ в этом групповом алфавите. С каждой парой $\langle A_i, B_i\rangle$ свяжем два *элементарных преобразования* слов в групповом алфавите $\mathcal{A} \cup \mathcal{A}^{-1}$ – переходы вида

$$UA_iV \longrightarrow UB_iV, \quad UB_iV \longrightarrow UA_iV,$$

где U и V – произвольные слова в групповом алфавите.

К этим элементарным преобразованиям добавим так называемые *тривиальные элементарные преобразования* слов в групповом алфавите $\mathcal{A} \cup \mathcal{A}^{-1}$ – переходы вида

$$Ua_i^{\varepsilon}a_i^{-\varepsilon}V \longrightarrow UV, \quad UV \longrightarrow Ua_i^{\varepsilon}a_i^{-\varepsilon}V,$$

где U и V – произвольные слова в групповом алфавите, а $\varepsilon \in \{-1, 1\}$. Преобразования первого вида называются *сокращениями*, а второго – *вставками*.

Саму упорядоченную пару слов $\langle A_i, B_i\rangle$ традиционно обозначают в виде $A_i = B_i$. Полученный объект обозначается в виде

$$\langle\langle\, a_1, \ldots, a_n \,|\, A_1 = B_1, \ldots, A_m = B_m \,\rangle\rangle,$$

и он будет служить заданием некоторой группы, которая будет обозначаться тем же способом. При этом $a_1, \ldots, a_n$ называются *образующими элементами* этой группы, а $A_1 = B_1, \ldots, A_m = B_m$ – ее *определяющими соотношениями*.

Для построения этой группы введем отношение эквивалентности $\sim$ на множестве всех слов в групповом алфавите.

Слова U и W в групповом алфавите назовем *эквивалентными* (обозначается $W \sim U$), если существует последовательность элементарных преобразований

$$W = W_0 \to W_1 \to \ldots \to W_k \to W_{k+1} \to \ldots \to W_s = U,$$

переводящая слово U в слово W.

Нетрудно показать, что отношение $\sim$ в рассматриваемом случае является отношением эквивалентности, т. е. оно рефлексивно, транзитивно и симметрично. Соответствующие классы эквивалентности будем обозначать через $[W]$.

На множестве классов эквивалентности естественным образом определяется умножение равенством

$$[W] \cdot [U] = [WU],$$

где WU – обычное произведение (сочленение, конкатенация) слов W и U.

Нетрудно проверить, что множество классов эквивалентности относительно введенной операции умножения является группой. При этом роль нейтрального элемента выполняет класс $[1]$, где через 1 обозначено пустое слово, а элементом, обратным к $[W]$, является $[\overline{W}]$, где

$$\overline{a_{i_1}^{\varepsilon_1} \dots a_{i_t}^{\varepsilon_t}} = a_{i_t}^{-\varepsilon_t} \dots a_{i_1}^{-\varepsilon_1}.$$

Построенная группа обозначается через

$$\langle\langle\, a_1, \dots, a_n \,|\, A_1 = B_1, \dots, A_m = B_m \,\rangle\rangle$$

и называется *группой, заданной образующими элементами* $a_1, \dots, a_n$ и *определяющими соотношениями* $A_1 = B_1, \dots, A_m = B_m$.

Если некоторая группа G изоморфна построенной группе, то говорят, что группа G имеет задание (генетический код, копредставление)

$$\langle\langle\, a_1, \dots, a_n \,|\, A_1 = B_1, \dots, A_m = B_m \,\rangle\rangle.$$

Например, симметрическая группа S_3 имеет задание

$$\langle\langle\, a, b \,|\, a^3 = 1,\, b^2 = 1, ba = a^2 b \,\rangle\rangle.$$

Группа $SL(2, Z)$ целочисленных матриц второго порядка с определителем, равным единице, имеет задание

$$\langle\langle\, a, b \,|\, a^6 = 1,\, b^4 = 1, a^3 = b^2 \,\rangle\rangle,$$

а ее факторгруппа по центру $PSL(2, Z)$ (проективная специальная целочисленная группа матриц второго порядка) имеет задание

$$\langle\langle\, a, b \,|\, a^3 = 1,\, b^2 = 1 \,\rangle\rangle.$$

Заметим, что группа кос на трех нитях и группа узла клеверный лист (трилистник) имеют одно и то же задание

$$\langle\langle\, a, b \,|\, a^3 = b^2 \,\rangle\rangle.$$

В связи с рассмотренным способом задания групп М. Дэн в работе 1911 года [11] сформулировал три алгоритмические проблемы, получившие название *фундаментальные проблемы М. Дэна*, – ***проблему тождества***, ***проблему сопряженности*** и ***проблему изоморфизма***.

Проблема тождества. *Требуется разработать общий метод (алгоритм), позволяющий по любому заданию группы*

$$\langle\langle\, a_1, \ldots, a_n \,|\, A_1 = B_1, \ldots, A_m = B_m \,\rangle\rangle$$

и по любым двум (групповым) словам W и U в этих образующих определить, равны ли элементы $[W]$ и $[U]$, т.е. можно ли из слова W вывести слово U, пользуясь указанными определяющими соотношениями и тривиальными соотношениями.

Проблема сопряженности. *Требуется разработать общий метод (алгоритм), позволяющий по любому заданию группы*

$$\langle\langle\, a_1, \ldots, a_n \,|\, A_1 = B_1, \ldots, A_m = B_m \,\rangle\rangle$$

и по любым двум (групповым) словам W и U в этих образующих определить, сопряжены ли элементы $[W]$ и $[U]$, т. е. найдется ли такое слово Z, что $[Z]^{-1}[W][Z] = [U]$ (можно ли из слова $\overline{Z}WZ$ вывести слово U, пользуясь указанными определяющими соотношениями и тривиальными соотношениями).

Проблема изоморфизма. *Требуется разработать общий метод (алгоритм), позволяющий по любым двум заданиям*

$$\langle\langle\, a_1, \ldots, a_n \,|\, A_1 = B_1, \ldots, A_m = B_m \,\rangle\rangle$$

и

$$\langle\langle\, b_1, \ldots, b_p \,|\, C_1 = D_1, \ldots, C_q = D_q \,\rangle\rangle$$

определить, будут ли изоморфны соответствующие группы.

Первые две проблемы были сформулированы М. Дэном в предыдущей работе 1910 года, а третью проблему можно обнаружить в работе Х. Титце 1908 года [9], но не выделенную там специально. Однако особое внимание этим проблемам было уделено именно в работе М. Дэна 1911 года [11], которая начинается с формулировки этих трех проблем.

Как уже отмечалось выше, в большой работе 1895 года «Analysis Situs» А. Пуанкаре ввел понятие фундаментальной группы, а в столь же большой работе 1908 года Х. Титце установил [9], что фундаментальные группы некоторых многообразий, заданных клеточными комплексами, имеют конечные задания. Начиная с работ Ж. Листинга 1848 года интенсивно изучаемый класс наглядных топологических объектов составляли узлы. В докдаде 1905 года В. Виртингер изложил метод нахождения группы узла по его проекции на евклидову плоскость. Ранее

М. Дэн предложил несколько иной способ нахождения задания группы узла, однако впоследствии метод В. Виртингера стал более распространенным. М. Дэн доказал, что *узел изотопически эквивалентен окружности тогда и только тогда, когда его группа абелева, а значит, циклическая.* Из сказанного можно понять, почему фундаментальные проблемы М. Дэна сразу привлекли внимание исследователей. Заметим лишь, что в проблемах М. Дэна речь шла о построении соответствующих *разрешающих алгоритмов.* Хотя по свидетельству ученика М. Дэна В. Магнуса позже М. Дэн допускал, что проблему равенства для всех конечно определенных групп, может быть, нельзя решить, как нельзя решить все математические задачи. И эти предположения позже блестяще подтвердились, когда было обнаружено, что некоторые знаменитые нерешенные математические проблемы в том или ином смысле могут быть сведены к проблемам изоморфизма и тождества для конечно определенных групп! А значит, проблемы изоморфизма и тождества для конечно определенных групп не проще, чем некоторые знаменитые нерешенные математические проблемы из «классических» разделов математики!

Почти полвека проблемы М. Дэна не поддавались решению – только в начале 50-х годов XX века Петр Сергеевич Новиков доказал [12], [13], [14], что искомые алгоритмы невозможно построить. Это позволило В. Магнусу выразить мысль, что каждый случай положительного решения той или иной алгоритмической проблемы можно рассматривать как триумф человека над природой.

Группа тривиального узла является бесконечной циклической, т. е. имеет задание

$$\langle\langle\, a \,|\, \emptyset \,\rangle\rangle,$$

а группа трилистника имеет задание

$$\langle\langle\, a, b \,|\, a^3 = b^2 \,\rangle\rangle.$$

Но как доказать, что эти группы не изоморфны?

Среди гомоморфных образов группы трилистника есть симметрическая группа $S(3)$ степени 3, имеющая задание

$$\langle\langle\, a, b \,|\, a^3 = 1,\, b^2 = 1, ba^2 = ab \,\rangle\rangle.$$

А симметрическая группа $S(3)$ нециклическая, значит, нециклическая и группа трилистника (гомоморфный образ циклической группы сам является циклической группой). Поэтому группа трилистника не изоморфна группе тривиального узла. Значит, ***трилистник нельзя развязать***!

А теперь мы отойдем от топологии узлов и кос и рассмотрим некоторые вопросы теории чисел, казалось бы весьма далекие от узлов и кос, но, как будет показано ниже, все же с ними достаточно тесно связанные.

Распределение простых чисел и гипотеза Римана

Вопрос о распределении простых чисел среди натуральных изучался в математике со времен Древней Греции, а может быть и раньше. Хорошо известно, что еще Евклид доказал, что множество простых чисел бесконечно.

А зачем это понадобилось Евклиду, если все в нашем мире конечно?

Значительнаяя часть материала заимствована из Википедии, хотя ее можно найти и в книгах по теории чисел.

Напомним, что функция $\pi(x)$ – число простых чисел, не превосходящих x, изучалась еще Лежандром в XVIII веке. Базируясь на таблицах простых чисел, составленных Фенкелем и Вегой, в 1796 году Лежандр предложил приблизить (оценить, заменить) функцию $\pi(x)$ функцией

$$\frac{x}{\ln(x) - B}$$

и, используя метод наименьших квадратов, оценил $B \approx 1,08366$.

Гаусс в письме Энке в 1849 году пишет, что еще в 1792 – 1793 годах он заметил (эмпирически, изучая таблицы простых чисел), что плотность простых чисел «в среднем близка к величине, обратно пропорциональной логарифму». Гаусс считал, что лучшее приближение к функции $\pi(x)$ дает другая функция – интегральный логарифм

$$Li(x) = \int\limits_2^x \frac{1}{\ln t} dt.$$

Кто же из двух великих математиков был прав? Оба! Все зависит от того, как мы понимаем слова «функция $f(x)$ может служить хорошим приближением (заменой) функции $\pi(x)$». Лежандр понимал это в смысле наименьшего квадратичного уклонения для значений в фиксированом наборе точек (он работал с построенными к тому времени таблицами простых чисел и по ним оценивал качество приближения и добился замечательных результатов). Позже точка зрения несколько изменилась: высказывание «функция $f(x)$ может служить хорошим приближением (заменой) функции $\pi(x)$» стали понимать как асимптотическую эквивалентность $\backsim$:

$$f \backsim g \quad \Longleftrightarrow \quad \lim_{x \to +\infty} \frac{f(x)}{(x)} = 1.$$

Но тогда следует заменить, что для любой константы C справедливы эквивалентности:

$$\frac{x}{\ln(x) + C} \backsim \frac{x}{\ln(x)} \backsim \int\limits_2^x \frac{1}{\ln t} dt.$$

Асимптотический закон распределения простых чисел – теорема аналитической теории чисел, утверждающая, что функция $\pi(x)$ эквивалентна функции $\frac{x}{\ln(x)}$, а значит, и функции $Li(x)$.

Доказательство этой теоремы было достаточно долгим. Первый существенный вклад в создание этого доказательства внес великий русский математик П. Л. Чебышев, который в 1848 – 1850 годах доказал, что если M – это верхний предел отношения $\frac{\pi(x)}{x/\ln(x)}$ при x стремящемся к $+\infty$, а m – его нижний предел, то выполняются неравенства $0,92129 \leq n \leq M \leq 1,10555$, а *если существует предел отношения* $\frac{\pi(x)}{x/\ln(x)}$, *то он равен* 1, т. е. если $m = M$, то $m = M = 1$. Однако доказать существование предела этого отношения удалось лишь 1896 году одновременно и независимо Адамару и Валле-Пуссену. Они оба построили доказательства на базе теории функций комплексного переменного, развивая идеи работы Римана, который в 1859 году предложил рассмотреть введенную еще Эйлером дзета-функцию $\zeta(s)$ вещественного аргумента s как дзета-функцию $\zeta(z)$ комплексного аргумента z. Распределение простых чисел среди натуральных и вдруг дзета-функция $\zeta(z)$ комплексного аргумента z!

Эйлер в 1737 году определил дзета-функцию $\zeta(s)$ вещественного аргумента s при $s > 1$ равенством

$$\zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s}$$

и доказал выполнимость равенства, носящего теперь название ***тождество Эйлера***:

$$\zeta(s) = \prod_{p\,-\,простое\ число} \frac{1}{1-p^{-s}}.$$

Тождество Эйлера дает разложение дзета-функции в бесконечное произведение и выражает одно из важнейших свойств дзета-функции.

Хорошо известны равенства $\zeta(2) = \pi^2/6$, $\zeta(4) = \pi^4/90$, т. е. равенства

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \pi^2/6, \quad \sum_{n=1}^{+\infty} \frac{1}{n^4} = \pi^4/90.$$

Более того, известны формулы, выражающие значения дзета-функции для четных натуральных чисел

$$2\zeta(2n) = (-1)^{n+1} \frac{(2\pi)^{2n}}{(2n)!} B_{2n},$$

где B_{2n} – числа Бернулли.

Относительно значений дзета-функции для нечетных натуральных чисел известно мало, есть гипотеза, что они все иррациональны и даже

трансцендентны, но большим успехом было доказательство в 1978 году Роже Апери иррациональности числа $\zeta(3)$.

Дзета-функция рассматривалась в работах Дирихле и Чебышева, но новый взгляд на дзета-функцию предложил Риман, который в своем знаменитом мемуаре 1859 года перешел от вещественного аргумента s к комплесному аргументу $s = \sigma + i\tau$, положив:

$$\zeta(\sigma + i\tau) = \sum_{n=1}^{+\infty} \frac{1}{n^{\sigma+i\tau}}.$$

Последний ряд, как знает каждый студент второго курса математического факультета, сходится при $\sigma > 1$.

Риман установил выполнимость ряда равенств для дзета-функции $\zeta(s)$ при $s = \sigma + i\tau$ и $\sigma = Res > 1$:

обратимость

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{+\infty} \frac{\mu(n)}{n^s},$$

где $\mu(n)$ – функция Мебиуса,

$$\zeta^2(s) = \sum_{n=1}^{+\infty} \frac{\tau(n)}{n^s},$$

где функция $\tau(n)$ – число делителей числа n,

$$\zeta^2(s)\zeta(2s) = \sum_{n=1}^{+\infty} \frac{2^{\nu(n)}}{n^s},$$

где функция $\nu(n)$ – число простых делителей числа n.

Функциональное уравнение Римана при s, отличном от 0 и 1:

$$\zeta(s) = 2^s \pi^{s-1} sin(\pi s/2)\Gamma(1-s)\zeta(1-s),$$

где $\Gamma(s)$ – гамма-функция Эйлера.

Для введенной Риманом функции $\xi(s) = 1/2\pi^{-s/2}s(s-1)\Gamma(s/2)\zeta(s)$, получившей название кси-функция Римана, функциональное уравнение принимает особенно простой вид

$$\xi(s) = \xi(1-s).$$

Все это позволило Риману продолжить дзета-функцию $\zeta(s)$, определенную при $Res > 1$ равенством

$$\zeta(s) = \sum_{n=1}^{+\infty} \frac{1}{n^s},$$

на всю комплексную плоскость, кроме точки $s = 1$, в которой у полученной аналитической дзета-функции $\zeta(s)$ простой полюс с вычетом, равным 1.

Из функционального уравнения Римана следует, что дзета-функция $\zeta(s)$ в полуплоскости $Res < 0$ имеет лишь простые нули – четные целые числа -2, -4, ..., $-n$, ..., которые называются «тривиальными» нулями дзета-функции.

Как показал Риман, особый интерес представляют «нетривиальные» нули дзета-функции. Они лежат в полосе $0 \leq Res \leq 1$, называемой ***критической полосой***, симметричны относительно вертикальной прямой $Res = 1/2$ и относительно вещественной оси, их мнимая часть отлична от нуля, т. е. они – чисто комплексные числа.

Гипотеза Римана *Находящиеся в критической полосе нули s дзета-функции* $\zeta(s)$ *лежат на прямой* $Res = 1/2$.

Гипотеза Римана включена Д. Гильбертом в 8-ю проблему из его знаменитого списка 23 математических проблем, получившего название «Проблемы Гильберта», и вошла четвертой в список из семи математических проблем – «Проблемы тысячелетия».

Почему же ***гипотеза Римана*** привлекла всеобщее внимание? Конечно, ***гипотеза Римана*** интересна сама по себе как вызов человеческому интеллекту, сформулированный человеческим интеллектом. Как доказал Х. Кох в 1901 году, ***гипотеза Римана*** эквивалентна выполнимости равенства

$$\pi(x) \;=\; Li(x) + O(\sqrt{x}\ln(x)),$$

которое дает возможность оценить ошибку, возникающую при замене функции $\pi(x)$ на функцию $Li(x)$.

В терминах пси-функции Чебышева

$$\psi(x) = \sum_{\substack{p^n \leq x \\ {}_{p-}\textit{простое}\ \textit{число} \\ {}_{n-}\textit{натуральное}\ \textit{число}}} \log p$$

асимптотический закон распределения простых чисел принимает особенно простой вид

$$\psi(x) \backsim x$$

Важность нулей дзета-функции Римана следует и из равенства Римана

$$\psi(x) \;=\; x \;-\; \sum_{\substack{s:\zeta(s)=0 \\ 0<Re(s)<1}} \frac{x^s}{s} \;-\; \log(2\pi) - \frac{1}{2}\log(1 - x^{-2}).$$

Последнее слагаемое $-\frac{1}{2}\log(1-x^{-2})$ отвечает тривиальным нулям дзета-функции s=-2, -4, ..., второе слагаемое

$$-\sum_{\substack{s:\zeta(s)=0\\0<Re(s)<1}}\frac{x^s}{s}$$

отвечает нетривиальным нулям s дзета-функции, т. е. нулям из критической полосы $0 < Res < 1$, а слагаемое $-\log(2\pi)$ отвечает полюсу x^s/s в нуле.

В 1949 году Эрдешом-Сельбергом было предложено «элементарное» доказательство асимптотического закона распределения простых чисел без выхода в комплексную плоскость, т. е. не использующее комплесного анализа. Мы здесь не будем обсуждать интересный вопрос «Какое доказательство проще: “неэлементарное” или “элементарное”?»'

Хорошо известно, что ***гипотеза Римана*** эквивалентна ряду утверждений, например, таким, взятым из статьи в Википедии:

при любом $x \geq 2657$ выполняется неравенство: $|\pi(x) - Li(x)| < \frac{1}{8\pi}\sqrt{x}\ln(x)$,

при любом $x \geq 73.2$ выполняется неравенство: $|\psi(x) - x| < \frac{1}{8\pi}\sqrt{x}\ln^2(x)$,

выполняется равенство:

$$\int_0^\infty \frac{(1-12t^2)}{(1+4t^2)^3}\int_{1/2}^\infty \log|(\sigma+it)|d\sigma dt = \frac{\pi(3-\gamma)}{32}.$$

Приведем один интересный, с нашей точки зрения, исторический факт, связанный с ***гипотезой Римана***. В одном из вариантов биографии Д. Гильберта описывается такой эпизод. В одном из выступлений Д. Гильберт сказал, что он рассчитывает дожить до того времени, когда будет доказана ***гипотеза Римана***, а самые молодые из присутствующих вполне могут дожить и до тех времен, когда будет доказана ***великая теорема Ферма***. Последний прогноз Д. Гильберта можно считать сбывшимся, хотя и не все с этим согласны, но вот первый его прогноз не оправдался. Даже гениям не всегда удавалось предсказать будущее! И еще один известный факт. На вопрос о том, что он сделает, если проснется через 500 лет, Д. Гильберт как будто бы ответил, что прежде всего спросит, доказана ли ***гипотеза Римана***.

Еще одно эквивалентное ***гипотезе Римана*** утверждение связано с диофантовыми уравнениями.

Диофантовы уравнения

А теперь мы расмотрим так называемые *диофантовы уравнения*, ведущие свою историю со времен Древней Грецции.

Под *диофантовым уравнением* понимается уравнение вида

$$F(x_1, \ldots, x_n) = 0,$$

где $F(x_1, \ldots, x_n)$ – полином с целыми коэффициентами от переменных $x_1,$ ldots, x_n при условии, что интересуются решениями этого уравнения в целых числах или в натуральных числах (это две разные, но тесно связанные между собой задачи).

С диофантовыми уравнениями связана ***10-я проблема Д. Гильберта*** – проблема, включенная Д. Гильбертом под номером 10 в знаменитый список из 23 проблем, сформулированных им на II Международном математическом конгрессе, состоявшемся в августе 1900 года в Париже.

10. Выяснение разрешимости произвольного диофантова уравнения. *Пусть задано диофантово уравнение с произвольным числом неизвестных и целыми рациональными коэффициентами; требуется указать способ, по которому с помощью конечного числа операций можно было бы узнать, разрешимо ли уравнение в целых рациональных числах или нет.*

«*Способ*», о котором идет речь в формулировке 10-й проблемы, теперь понимается как ***алгоритм***. Хорошо известно, какие трудности возникают при исследовании диофантовых уравнений, даже такого, казалось бы простого, как уравнение Пелля $x^2 - dy^2 = 1$. Поэтому со временем появились предположения, что искомого алгоритма просто не существует. Справедливость этих предположений была установлена во второй половине XX века в серии работ М. Дэвиса, Х. Путнам, Дж. Робинсон и Ю.В. Матиясевича [15], что явилось одним из выдающихся фундаментальных достижений теории алгоритмов второй половины XX века. Заметим, что в настоящее время неизвестно, *возможно ли построить алгоритм, позволяющий по произвольному уравнению вида*

$$F(x_1, \ldots, x_n) = 0,$$

где $F(x_1, \ldots, x_n)$ – полином с целыми коэффициентами от переменных $x_1, \ldots, x_n$, определить, имеет ли оно решение в рациональных числах. Интересно это сопоставить со следующим фактом: вопрос о разрешимости уравнений указанного вида в действительных числах алгоритмически разрешим, что было установлено А. Тарским на основе глубокого обобщения известного читателю из курса «Алгебра» метода Штурма, относящегося к уравнениям с одной неизвестной, т.е. когда $n = 1$, а вопрос о разрешимости уравнений этого вида в комплексных числах легко решается на основе так называемой основной теоремы о многочленах над полем комплексных чисел.

Есть ли "достаточно тесная" связь между "весьма далекими" на первый взгляд вопросами, обсуждавшимися выше? Оказывается есть: по

Гипотезе Римана Ю. В. Матиясевич [16] построил диофантово уравнение

$$F_{RH}(x_1, \ldots, x_n) = 0 \qquad (RH)$$

такое, что

Гипотеза Римана *верна тогда и только тогда, когда уравнение (RH) не имеет решения в натуральных числах.*

В ряде работ уравнение (RH) упрощалось и в настоящее время его можно, в принципе, выписать в явном виде (только надо не допустить ошибки при выписывании левой части этого уравнения). В ряде работ можно найти явный вид этого уравнения. А как проверить, не допущена ли ошибка (опечатка)?

Уравнение (RH) позволяет утверждать, что если ***гипотеза Римана*** ложна, то алгоритм, перебирающий в некотором порядке все наборы натуральных чисел $(a_1, \ldots, a_n)$ и проверяющий, выполняется ли равенство

$$F_{RH}(a_1, \ldots, a_n) = 0,$$

позволит установить ложность ***гипотезы Римана***. Если же ***гипотеза Римана*** верна, то указанный алгоритм никогда не закончит работу (не остановится, «зависнет»).

Арифметика Пеано

Рассмотрим простую, но очень важную по ряду причин формальную (аксиоматическую) теорию – ***Арифметику Пеано***, обозначаемую через ArP. Эта аксиоматическая теория базируется на языке Исчисления предикатов с равенством, сигнатура которого включает один индивидный константный символ 1, один одноместный функциональный символ s и два двуместных функциональных символа $+$ и $\cdot$. Двуместный предикатный символ $=$ относится к логическим символам. Логическими аксиомами являются стандартные логические аксиомы исчисления предикатов и дополнительные *аксиомы равенства*:

1) $(\forall x)x = x$,
2) $(\forall x)(\forall y)(x = y \to y = x)$,
3) $(\forall x)(\forall y)(\forall z)(x = y \,\&\, y = z) \to x = z)$,
4) $(\forall x)(\forall y)(x = y \to s(x) = s(y))$,
5) $(\forall x)(\forall y)(\forall z)(x = y \to (x + z = y + z \,\&\, z + x = z + x))$,
6) $(\forall x)(\forall y)(\forall z)(x = y \to (x \cdot z = y \cdot z \,\&\, z \cdot x = z \cdot x)$.

Нелогическими аксиомами теории ArP являются все формулы вида:

1) $(\forall x)\neg 1 = s(x)$,
2) $(\forall x)(\forall y)(s(x) = s(y) \to x = y)$,
3) $((A_x[1] \,\&\, (\forall x)(A \to A_x[s(x)]) \to (\forall x)\, A)$, где A – произвольная формула с одной свободной переменной x,
4) $(\forall x)(x + 1 = s(x))$,

5) $(\forall x)(\forall y)(x + s(y) = s(x + y))$,
6) $(\forall x)(x \cdot 1 = x)$,
7) $(\forall x)(\forall y)(x \cdot s(y) = x \cdot y + x)$.

Из предыдущего сразу следует, что если ***гипотеза Римана*** ложна, то это можно доказать даже в ***Арифметике Пеано*** ArP. Но при этом «кто-то внешний по отношению к ***Арифметике Пеано*** ArP должен *понять*, что доказана ложность ***гипотезы Римана***».

Зададим группу образующими элементами и определяющими соотношениями

$$G_{RH} = \langle\langle\, a, b \,|\, \{a^{\alpha_1} b^{\alpha_1} \dots a^{\alpha_n} b^{\alpha_n} = 1 \,|$$

$$\alpha_1, \dots, \alpha_n \in N \& F_{RH}(\alpha_1, \dots, \alpha_n) = 0 \,\}\, \rangle\rangle.$$

В таком случае:

Гипотеза Римана *верна тогда и только тогда, когда группа* G_{RH} *является свободной группой ранга* 2.

Защищенный документооборот

А теперь мы рассмотрим один подход к построению защищенного документооборота. Он объединен общим названием **RSA** и существенно использует простые числа.

Система **RSA**, разработанная тремя криптографами R. L. Rivest , A. Shamir и L. Adleman [17] и названная по первым буквам их фамилий, позволяет организовать криптографически защищенный документооборот между удаленными абонентами, подкрепленный Электронной ифровой подписью (ЭЦП). При этом, в определенном смысле, снимается одна из труднейших проблем – проблема распределения ключей шифрования.

Каждый абонент A криптографически защищенной системы документооборота на базе **RSA** выбирает пару различных простых чисел p_A и q_A. Они будут входить в состав его ***секретного ключа***. На выбираемые простые числа p_A и q_A накладываются некоторые ограничения с целью обеспечения достаточной криптографической стойкости рассматриваемого протокола, однако обсуждение этого вопроса не входит в наши цели. Просто считаем, что p_A и q_A – «достаточно большие» неравные простые числа, например, имеющие в двоичном представлении порядка 500 знаков.

Абонент A вычисляет число $n_A = p_A \cdot q_A$. Кроме того, абонент A выбирает «не очень большое» натуральное число e_A взаимно простое с $\varphi(n_A) = (p-1)(q-1)$. Требования к выбору числа e_A обсудим позже. В настоящее время рекомендуется в качестве e_A выбирать простые числа Ферма, т. е. простые числа вида

$$2^{2^n} + 1.$$

Это связано с тем, что двоичная запись чисел Ферма содержит лишь две 1, что существенно ускоряет возведение произвольного числа в эту степень.

Абонент A обращает элемент $[e_A]$ в кольце $Z(\varphi(n_A))$, т. е. находит такое натуральное число d_A, что

$$e_A \cdot d_A \equiv 1 (\mod \varphi(n_A)).$$

Такой выбор чисел в силу теоремы Л. Эйлера обеспечивает справедливость следующего утверждения

если натуральное число a взаимно просто с n_A, то

$$a^{e_A \cdot d_A} \equiv a (\mod n_A).$$

Выбор в качестве числа n_A произведения двух различных простых чисел позволяет отказаться от условия взаимной простоты a и n_A, т. е. справедливо следующее утверждение

для произвольного натурального числа a справедливо сравнение

$$a^{e_A \cdot d_A} \equiv a (\mod n_A). \qquad (BP)$$

В свободном доступе, например на соответствующем сайте, размещаются следующие данные об абоненте A:

$$A, \quad n_A, \quad e_A.$$

Эти данные составляют ***открытый ключ*** абонента A, а числа d_A, p_a и q_A составляют его ***секретный, закрытый, ключ***.

Абонент B для отправки абоненту A сообщения M в криптографически защищенной форме выполняет следующие действия.

1) Некоторым стандартным способом переводит его в числовой код a, например используя ASCII-кодирование. При этом должно выполняться неравенство $a < n_A$. Для этого исходное сообщение M разбивается на блоки фиксированной длины, преобразуется и передается поблочно.

2) Вычисляет

$$C = rem(a^{e_A}, n_A).$$

3) Отправляет C по имеющемуся каналу связи.

Абонент A, получив C, выполняет следующие действия.

1) Вычисляет

$$D = rem(C^{d_A}, n_A)$$

В силу утверждения (BP)

$$C^{d_A} \equiv a^{e_A \cdot d_A} \equiv a (\mod n_A).$$

Поэтому $D \equiv a(\mod n_A)$. А так как $1 \leq D < n_A$ и $1 \leq a < n_A$, то $D = a$.

2) По числу a абонент A стандартным способом восстанавливает исходное сообщение M.

Хэш-функция – это отображение h множества всех слов Σ^* в алфавите Σ во множество Δ_n^* слов фиксированной длины n в алфавите Δ. Обычно в качестве алфавита Δ выбирается двубуквенный алфавит $\{0, 1\}$.

Для слова (сообщения) w в алфавите Σ значение $h(w)$ называется *сверткой* (дайджестом) слова (сообщения) w. Сама хэш-функция h также часто называется *сверткой*.

На функцию h накладывается ряд требований с целью обеспечения ее криптографической стойкости, в частности, требуется, чтобы она была односторонней функцией, т. е. вычисление значения $h(w)$ по w должно быть «вычислительно простой задачей», а нахождение по v хотя бы одного w (при условии, что оно существует) такого, что $v = h(w)$ должно быть «вычислительно сложной задачей». Мы не будем утонять смысл слов «просто» и «сложно». Мы также не будем обсуждать другие требования, накладываемые на ***хэш-функции*** для обеспечения надежности рассматриваемого варианта ЭЦП.

Абонент B может подписать ЭЦП сообщение M. Для этого используется некоторая фиксированная хэш-функция h и считается, что для сообщения w и любого абонента C выполняется неравенство $h(w) < n_C$.

Абонент B, кроме вычисленного указанным выше способом числа C, вычисляет S по формуле

$$S = rem(h(C)^{d_B}, n_B),$$

где d_B – ***секретный ключ*** абонента B, и отправляет абоненту A

$$B, \quad C, \quad S.$$

Абонент A описанным выше способом восстанавливает сообщение M.

Для проверки авторства полученного сообщения абонент A берет в открытом доступе ***открытый ключ*** e_B абонента B и выполняет следующие действия.

1) Вычисляет

$$C' = rem(S^{e_B}, n_B).$$

2) Если $C' \neq h(C)$, то сообщение не рассматривается как отправленное абонентом B в силу того, что

$$S^{e_B} \equiv h(C)^{d_B \cdot e_B} \equiv h(C)(\mod n_B).$$

А так как $h(C) < n_B$, то должно выполняться равенство

$$rem(S^{e_B}, n_B) = h(C).$$

3) Если $C' = h(C)$, то сообщение рассматривается как отправленное абонентом B.

Для обмена сообщениями, зашифрованными в соответствии с действующим в РФ стандартом ГОСТ 28147-89 или действующем в США стандартом AES, абонентам A и B требуется общий ключ K для зашифрования-расшифрования (симметричные системы шифрования). Обеспечение абонентов ключами K требует создания целой структуры, обеспечивающей формирование «стойких» ключей, их распределение, замену отработанных ключей на новые, уничтожение старых или их надежное хранение при долговременном шифровании и т. д. Но, как заметили Диффи и Хеллман (Diffie, Hellman), в некоторых случаях можно обойтись «собственными силами».

Диффи и Хеллман (Diffie, Hellman) в работе [18] предложили следующий протокол формирования общего ключа.

Абоненты A и B заранее или перед началом обмена шифрсообщениями договариваются по открытому каналу связи о выборе общего конечного поля F, порождающего элемента g мультипликативной группы F^* этого поля, и о стандартной (канонической) форме $NF(f)$ представления элементов f поля F. Напомним, что если $GF(q)$ – конечное поле из q элементов, $p(x)$ – неприводимый полином степени n над этим полем, $GF(q_1)$ – конечное поле из q_1 элементов, $p_1(x)$ – неприводимый полином степени n_1 над этим полем и $q^n = q_1^{n_1}$, то конечные поля $GF(q)[x]/(p(x))$ и $GF(q_1)[x]/(p_1(x))$ состоят из одного и того же числа $q^n = q_1^{n_1}$ элементов и изоморфны, однако нормальные формы элементов этих полей могут существенно различаться.

Протокол выработки общего ключа:

абонент A "случайным образом" выбирает натуральное число α, вычисляет g^α, $g_A = NF(g^\alpha)$ и передает абоненту B по открытому каналу связи элемент g_A;

параллельно и независимо абонент B «случайным образом» выбирает натуральное число β, вычисляет g^β, $g_B = NF(g^\beta)$ и передает абоненту A по открытому каналу связи элемент g_B;

абонент A вычисляет g_B^α и $K_A = NF(g_B^\alpha)$;

абонент B вычисляет g_A^β и $K_B = NF(g_A^\beta)$;

общий ключ $K = K_A = K_B$.

В. М. Сидельников в работе [19] предложил более общий алгебраический подход к реализации протокола Диффи – Хэллмана выработки общего ключа с использованием подходящих полугрупп. Одна из реализаций этого подхода В. М. Сидельникова была предложена в работе [2] и вызвала определенный интерес. Она базировалась на группах ко $B(n)$.

$$B(n) = \langle\langle\, \sigma_1, \ldots, \sigma_{n-1} \,|\, \{\, \sigma_i\sigma_j = \sigma_j\sigma_i \,|\, |i-j| > 1 \,\} \cup \{\, \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \,|\, 1 \le i < n \,\} \,\rangle\rangle$$

Хорошо известно, что центр группы $B(n)$ является циклической группой, порожденной элементом

$$\Delta = (\sigma_1\sigma_2\dots\sigma_{n-1})(\sigma_1\sigma_2\dots\sigma_{n-2}\dots)(\sigma_1\sigma_2)\sigma_1,$$

который называется элементом Гарсайда.

Гарсайд установил, что любой элемент W группы кос однозначно представим в виде $\Delta^m\Omega$, где m – целое число, а Ω – положительный элемент (коса), т. е. слово вида $\sigma_{i_1}\sigma_{i_2}\dots\sigma_{i_t}$. Это представление называется *нормальной формой* элемента W и обозначается через $NF(W)$.

В группе $B(2(n+1))$ рассматриваются подгруппы

$$L(n) = gr(\sigma_1,\dots,\sigma_n) \quad \text{и} \quad R(n) = gr(\sigma_{n+2},\dots,\sigma_{2n+1}).$$

Если $g \in L(n)$, а $h \in R(n)$, то $gh = hg$, т. е. элементы этих подгрупп коммутируют.

Абоненты A и B выбирают (заранее или в начале сеанса связи) неединичный элемент G группы $B(n)$.

Для выработки общего ключа абонент A выбирает $L_A \in L(n)$ и $R_A \in R(n)$, вычисляет элемент L_AGR_A, его нормальную форму $W_A = NF(L_AGR_A)$ и отправляет элемент W_A абоненту B.

Абонент B выбирает $L_B \in L(n)$ и $R_B \in R(n)$, вычисляет элемент R_BGL_B, его нормальную форму $W_B = NF(R_BGL_B)$ и отправляет элемент W_B абоненту A.

Абонент A вычисляет $K_A = NF(L_AW_BR_A)$, а абонент B вычисляет $K_B = NF(R_BW_AL_B)$. Так как

$$L_AW_BR_A = L_A(R_BGL_B)R_A = R_B(L_AGR_A)L_B = R_BW_AL_B,$$

то $K_A = K_B$. Общий ключ, полученный по этому протоколу, – это

$$K = K_A = K_B.$$

Уравнения в свободных группах

Обозначим через F_2 свободную группу ранга 2 со свободными образующими a и b. Это один из простейших алгебраических объектов – элементами группы F_2 служат слова в алфавите $\{a, a^{-1}, b, b^{-1}\}$, не содержащие подслов вида $a^\varepsilon a^{-\varepsilon}$ и $b^\varepsilon b^{-\varepsilon}$, где $\varepsilon \in \{-1, 1\}$. Такие слова называются *несократимыми*. Замена слов вида $Ua^\varepsilon a^{-\varepsilon}V$ и $Ub^\varepsilon b^{-\varepsilon}V$ на слово UV называется сокращением. Чтобы перемножить в группе F_2 два элемента W и U (два несократимых слова), надо образовать слово WU (конкатенация слов W и U) и, выполнив все возможные сокращения, получить из слова WU несократимое слово $\overline{WU}$. Несократимое слово $\overline{WU}$ и считается произведением несократимых слов W и U. Правда,

есть еще более простой алгебраический объект – свободная полугруппа S_2 ранга 2 со свободными образующими a и b. Ее элементами служат слова в алфавите $\{a, b\}$. Чтобы перемножить в полугруппе S_2 два элемента W и U, надо образовать слово WU (конкатенация слов W и U). Но ее мы пока не будем рассматривать.

Зафиксируем счетный алфавит X неизвестных

$$\{x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}, \ldots\}.$$

Уравнение с n неизвестными в свободной группе F_2 – это равенство вида

$$w(x_1, \ldots, x_n, a, b) = u(x_1, \ldots, x_n, a, b),$$

где $w(x_1, \ldots, x_n, a, b)$ и $u(x_1, \ldots, x_n, a, b)$ – несократимые слова в объединенном алфавите

$$\{a, a^{-1}, b, b^{-1}\} \cup \{x_1, x_1^{-1}, \ldots, x_n, x_n^{-1}\}.$$

Решением этого уравнения в свободной группе F_2 называется любой такой набор g_1, ..., g_n элементов этой группы, для которого верно равенство

$$w(g_1, \ldots, g_n, a, b) = u(g_1, \ldots, g_n, a, b),$$

в группе F_2.

Базируясь на обсуждавшихся выше результатах Ю. В. Матиясевича [16], мы можем построить такое разрешенное относительно неизвестных уравнение

$$w(x_1, \ldots, x_n) = [a, b], \qquad (RH)$$

где $[a, b] = aba^{-1}b^{-1}$ – коммутатор элементов a и b, что

гипотеза Римана *справедлива тогда и только тогда, когда уравнение* (RH) *не имеет в свободной группе* F_2 *такого решения* g_1, ..., g_n, что g_1, ..., g_t *лежат в коммутанте* $F_2^{(1)}$ *свободной группы* F_2 (t – подходящее фиксированное число).

Эпилог или пролог – окончание или начало?

«Все переплелось» – скажет читатель: простые числа, гипотеза Римана, диофантовы уравнения, уравнения в свободных группах, узлы и косы, фундаментальные группы, шифрование и защита информации!

О чем же эта заметка? Для кого она написана, кому, по мнению автора, могла бы принести некоторую пользу? Прежде всего рекомендуем читателю вернуться к эпиграфу – может быть тогда вопрос отпадет. Кроме того, автору хотелось бы, чтобы эта заметка не причинила вреда! Поэтому он старался не перегружать ее математическими определениями, а тем более доказательствами. Она ориентирована на

студента, может быть искренне заблуждающегося по поводу ценности тех или иных математических фактов из-за своей малой математической, да и общей образованности, что с годами может и пройти, а может и не пройти, а не на студента, не принимающего знания по «идейным убеждениям». Может быть, первый увидит и поймет, что история упорно свидетельствует: весьма трудно сегодня знать, что нам потребуется завтра. А вывод? Впитывай знания, как губка впитывает воду! Может быть потом утолишь жажду! Используй выпавшую тебе удачу – поучиться у знающих людей, другой такой возможности тебе может быть и не представится.

И еще одно замечание по поводу «практической ценности» математических знаний. У начинающих изучение математики студентов нередко возникает вопрос о «практическом применении» даже первоначальных сообщаемых им математических фактов, которые к тому же и не слишком фундаметальны, но, может быть, кажутся таковыми слушателям. Великий английский математик Г. Х. Харди пишет: «Я никогда не делал ничего “полезного”. Ни одно из моих открытий не произвело и не имеет шансов произвести, будь то явным или неявным образом, к добру или ко злу, ни малейшей перемены в удобствах жизни... При оценке по стандартам практики значение моей математической жизни равно нулю.» Г. Х. Харди «гордился» тем, что математические проблемы, которые он исследовал, относятся к «чистой математике» и никогда не найдут никакого практического применения. И прежде всего к «чистой математике» он относил «Теорию чисел». Можно предположить, как был бы огорчен великий Г. Х. Харди, узнав, насколько интенсивно и успешно используется его любимая «чистая теория чисел» в современной «прикладной» криптографии для решения весьма практических и не всегда «чистых» задач, возникающих в реальной жизни. Может быть, его бы утешило то, что, с другой стороны, некоторые направления «прикладной математики», созданные для решения прикладных задач, не только не смогли их решить, но и сами «засохли». А «чистая» математика продолжает расти, цвести и приносить плоды, в том числе и «практические».

Ж. Адамар пишет [20]: «Практическое приложение обнаруживается, когда его не ищут, и можно сказать, что весь прогресс человечества зиждется на этом принципе... Практические вопросы чаще всего удается разрешить с помощью существующих теорий... Редко случается так, что важные математические изыскания предпринимаются непосредственно ввиду той или иной практической пользы; мотивировкой их является то же стремление, которое служит основой всякой научной деятельности, – стремление узнать и понять.»

Ссылки

1. Listing J. B. *Vorstudien zur Topologie.* Göttingen : Vandenhoeck und Ruprecht, 1848.

2. Ko, K. H., Lee S .J., Cheon J. H., Han J. W., Kang J. S., Park C. *New public-key cryptosystem using braid groups* // Advances in Cryptology – Crypto'2000, LNCS. Springer, Berlin, 2000. Vol. 1880. P. 166–183.

3. Artin E. *Theorie der Zöpfe.* // Abh. Math. Sem. Hamburg. Univ. 1925. Bd. 4. S. 47 – 72.

4. Марков А. А. *Основы алгебраической теория кос* // Труды МИАН им. В. А. Стеклова, 1945. Т. 16.

5. Марков А.А. *Неразрешимость проблемы гомеоморфии* // ДАН СССР. 1958. Т. 121, № 2. С. 218 – 220.

6. Thompson A. *Thin position and the recognotion problem for* S^3 // Mathematical Research Letters. 1994. Vol. 1. P. 613–630.

7. Матвеев С. В. *Алгоритм распознавания трехмерной сферы (по А. Томпсон)* // Матем. сборник. 1995. Т. 186, № 5. С. 69 – 84.

8. Володин И. А., Кузнецов В. Е., Фоменко А. Т. *О проблеме алгоритмического распознавания стандартной трехмерной сферы* // Успехи матем. наук. 1974. Т. 29, № 5. С. 71 – 168.

9. Tietze H. *Über topologischen Invarianten mehrdimensionaler Mannigfaltigkeiten* // Monatsh. Math. Phys. 1908. Vol. 19. P. 1 – 118.

10. Wirtinger W. *Über die Verzweigungen bei Funktionen von zwei Veränderlichen* // Jahresbericht der Deutschen Mathematiker-Vereinigung. 1905. Bd. 14. S. 517.

11. Dehn M. *Über unendliche diskontinuerliche Gruppen* // Math. Ann. 1911. Bd. 71. S. 116 – 144.

12. Новиков П. С. *Об алгоритмической неразрешимости проблемы тождества теории групп* // ДАН СССР. 1952. Т. 85, № 4. С. 709 – 712.

13. Новиков П. С. *Об алгоритмической неразрешимости проблемы тождества слов в теории групп* // Труды МИАН. Т. 44. М. : Наука, 1955.

14. Адян С. И., Дурнев В. Г. *Алгоритмические проблемы для групп и полугрупп* // Успехи матем. наук. 2000. Т. 55, № 2. С. 3 – 94.

15. Матиясевич Ю. В. *Диофантовость перечислимых множеств* // ДАН СССР. 1970. Т. 130, № 3. С. 495 – 498.

16. Матиясевич Ю. В. *Десятая проблема Гильберта.* М. : Наука, 1993.

17. Rivest R. L., Shamir A., Adleman L. *A method for obtaining digital signatures and public-key cryptosistems* // Comm. ACM. 1978. Vol. 21, №2. P. 120 – 126.

18. Diffie W., Hellman M. E. *New directions in cryptography* // IEEE Transaction Information Theory. 1976. Vol. 22, №6. P. 644 – 654.

19. Сидельников В. М., Черепнев М. А., Ященко В. В. *Системы открытого распределения ключей на основе некоммутативных полурупп* // Докл. РАН. 1993. Т. 332, №5. С. 566 – 567.

20. Адамар Ж. *Исследование психологии процесса изобретения в математике* М. : Советское радио, 1970.

УДК 517.51

И. П. Иродова
Ярославский государственный университет им. П. Г. Демидова
E-mail: IrinaIrodova@gmail.com

О многочлене наилучшего приближения

Описаны способы нахождения многочлена наилучшего приближения в пространствах $C[a,b]$ и $L_2,\rho[a,b]$.

Библиография: 2 названия

Ключевые слова: многочлен, приближение, теорема Чебышева.

Классическая задача теории приближения связана с нахождением многочлена наилучшего приближения в некотором банаховом пространстве X. Обозначим через P_{n-1} пространство многочленов степени не более $n-1$, а через $e(f,P_{n-1})_X$ – наилучшее приближение функции $f\in X$ многочленами пространства P_{n-1}. Известно, что эта задача может быть решена в пространстве $X=L_2[a,b]$ (см., например, [1]). В этом случае для нахождения многочлена наилучшего приближения $p^*\in P_{n-1}$ нужно решить систему линейных уравнений. Чтобы сформулировать соответствующий результат, выберем базис пространства P_{n-1}, состоящий из многочленов $p_1,...,p_n$. Тогда $p^*=a_1^*p_1+...+a_n^*p_n$. А так как функция $f-p^*$ ортогональна пространству P_{n-1}, то $(f-p^*,p_i)=0$, $i=1,...,n$. Получаем систему n уравнений с n неизвестными:

$$(f,p_i)=\sum_{j=1}^{n}a_j^*(p_j,p_i),\ i=1,...,n. \tag{1}$$

Здесь, как обычно, $(g,\psi)=\int_a^b g(t)\psi(t)dt$.

Определитель этой системы, который называется определителем Грама, отличен от нуля, а значит, система имеет единственное решение. Однако, если базис выбран неудачно, определитель системы может быть близок к нулю. В этом случае система является плохо обусловленной. Самый простой способ избежать возникающих при этом вычислительных погрешностей – выбрать ортонормированный базис. Тогда $a_i^*=\ (f,p_i)$.

Можно обобщить эту задачу, выбрав в качестве X пространство $L_2, \rho[a,b]$, где $\rho > 0$ – весовая функция. Система (1) не изменится, нужно только учесть, что в этом случае $(g,\psi) = \int_a^b g(t)\psi(t)\rho(t)dt$.

Пусть теперь $X = C[a,b]$. Так как пространство $C[a,b]$ не является гильбертовым, то сейчас нельзя задачу нахождения многочлена наилучшего приближения свести к задаче решения системы линейных уравнений. Есть только отдельные классы функций, когда эта задача все-таки имеет решение. Например, если $f(t) = t^n$, то многочлен наилучшего приближения $p^*(t) = t^n - \overline{T_n}(t)$, где $\overline{T_n}$ – многочлен Чебышева степени n, старший коэффициент которого равен единице, то есть даже для решения такой частной задачи потребовалось разработать специальный аппарат, связанный с многочленами Чебышева (см., например, [2]). В общем случае задача нахождения многочлена наилучшего приближения в пространстве $C[a,b]$ решается приближенно с помощью алгоритма Ремеза. Этот алгоритм построен на свойстве многочлена наилучшего приближения, сформулированном в следующей теореме.

Теорема (Чебышев). *Многочлен $p^* \in P_{n-1}$ является многочленом наилучшего приближения для функции f из пространства $C[a,b]$ тогда и только тогда, когда существуют $n+1$ точки $\{t_i\}_{i=1}^{n+1}$, $a \le t_1 < t_2 < ... < t_{n+1} \le b$, в которых разность $\Delta(t) = f(t) - p^*(t)$ принимает значение $\pm\|\Delta\|_{C[a,b]}$ и $\Delta(t_{i+1}) = -\Delta(t_i)$.*

Алгоритм Ремеза позволяет за конечное число шагов найти многочлен, который мало отличается от многочлена наилучшего приближения. Хотя алгоритм показывает хорошие результаты, но он применим для нахождения многочленов не очень высоких степеней. Чаще всего на практике строят интерполяционные многочлены, а не многочлены наилучшего приближения.

Известно, что $\|f\|_{C[a,b]} = \lim\limits_{p\to\infty} \|f\|_{L_p[a,b]}$. Алгоритм, который описан ниже, предлагает перенести метод, который используется для вычисления наилучшего приближения в пространстве $L_2, \rho[a,b]$ на пространство $C[a,b]$. Идея использования среднеквадратического приближения для нахождения наилучшего приближения в пространстве непрерывных функций путем изменения весовой функции принадлежит Р. В. Хеммингу (1968 г).

Положим $k = 0, \rho_0 = 1$. На первом шаге найдем многочлен p_{k+1}, решая задачу

$$\|f - p_{k+1}\|_{L_2,\rho_k[a,b]} = \min_{p\in P_{n-1}} \|f - p\|_{L_2,\rho_k[a,b],}.$$

Затем положим $k := k+1$, вычислим весовую функцию

$$\rho_k(t) = \left(f(t) - p_k(t)\right)^{2k}$$

и перейдем к первому шагу.

Обозначим $\varepsilon_{k+1} = \|f - p_{k+1}\|_{L_2,\rho_k[a,b]}$. Тогда

$$e(f, P_{n-1})_{C[a,b]} \approx \lim_{k\to\infty} (\varepsilon_k)^{1/k}.$$

Заметим, что этот алгоритм не имеет строгого математического обоснования, его нельзя даже назвать эвристическим, так как неизвестно, дает ли он хорошие результаты. Единственным обоснованием алгоритма может служить результат, доказанный Джексоном, а именно

$$\lim_{p\to\infty} e(f, P_{n-1})_{L_p[a,b]} = e(f, P_{n-1})_{C[a,b]}.$$

Таким образом, этот алгоритм нуждается в дополнительных исследованиях, что можно предложить сделать студентам на специальных курсах по теории приближения.

Ссылки

1. *Колмогоров А. Н., Фомин С. В.* Элементы теории функций и функционального анализа. М. : Наука, 1976.

2. *Бердышев В. И., Петрак Л. В.* Аппроксимация фунций, сжатие численной информации, приложения. Ектеринбург : УрО РАН, 1999.

УДК 517.521

В. С. Климов
Ярославский государственный университет им. П. Г. Демидова
E-mail: klimov@uniyar.ac.ru

Задачи на тему «Последовательности с ограниченным изменением»

В работе обсуждается методика изложения темы «Последовательности с ограниченным изменением» в курсе математического анализа. Приводится набор задач, посвящённых данной теме.

Библиография: 1 название.

Ключевые слова: числовая последовательность, последовательность с ограниченным изменением, числовой ряд, сходимость.

Последовательностям с ограниченым изменением сильно не повезло. Их значение в теории рядов была выяснено в работах Дедекинда и Харди, выполненных более века назад. Однако в литературе по математическому анализу им уделяется крайне мало места (как, например, в задачнике [1]), чаще всего они и вовсе не упоминаются.

Автор полагает, что знакомство с этим кругом вопросов весьма желательно для студентов, интересующихся математическим анализом. Решение соответствующих задач могло бы активизировать освоение теории рядов и составить (при определённом развитии) тему курсовой или дипломной работы.

Приведу некоторые определения и обозначения. Пусть $\mathbb{C}$ – поле комплексных чисел, $z_k \in \mathbb{C}, (k = 1, 2, \cdots)$, p и q – натуральные числа, причём $p < q$. Число

$$\bigvee_p^q z_k := \sum_{k=p+1}^{q} |z_k - z_{k-1}|$$

будем называть изменением последовательности $\{z_k\}$ на промежутке $[p, q]$. Оно совпадает с длиной ломаной в вершинах $z_p, z_{p+1}, \cdots, z_q$.

Непосредственно из определения вытекает возрастание последовательности $t_n = \overset{n}{\underset{1}{V}} z_k$, $(n = 2, 3, \ldots)$.. Если последовательность t_n сходится, то её предел называется полным изменением (вариацией) последовательности $\{z_k\}$ и обозначается символом $\overset{\infty}{\underset{1}{V}} z_k$. Совокупность последовательностей $\{z_k\}$ с конечным полным изменением обозначим символом BV. Класс BV заведомо не пуст. Очевидно следующее утверждение: для того чтобы числовой ряд

$$\sum_1^\infty w_k$$

абсолютно сходился, необходимо и достаточно, чтобы последовательность $s_n = w_1 + \ldots + w_n$ его частичных сумм принадлежала классу BV.

Представленные ниже задачи существенно различаются по степени трудности. Большинство задач относительно просто, однако имеются задачи повышенной трудности: они отмечены звёздочкой.

1. Для того чтобы монотонная последовательность $z_k \in \mathbb{R}$ входила в класс BV, необходимо и достаточно, чтобы она была ограниченной, при этом

$$\overset{q}{\underset{p}{V}} z_k = |z_p - z_q|, \qquad \overset{\infty}{\underset{p}{V}} z_k = |z_\infty - z_1|, \text{где } z_\infty = \lim_{k\to\infty} z_k.$$

2. Сумма (разность, произведение) двух последовательностей класса BV есть последовательность того же класса.

3. Если $\{z_k\}, \{w_k\}$ – последовательности класса BV и $|w_k| > m > 0 \forall k$, то их частное z_k/w_k также принадлежит этому классу.

4. Последовательность $\{z_k\}$ имеет ограниченное изменение в том и только том случае, когда её действительная и мнимая части имеют ограниченное изменение.

5. Для того чтобы действительная последовательность $\{x_k\}$ принадлежала классу BV, необходимо и достаточно, чтобы она представлялась в виде разности двух возрастающих и ограниченных последовательностей.

6. Комплексная последовательность $\{z_k\}$ принадлежит классу BV в том и только том случае, если найдутся четыре возрастающие и ограниченные последовательности $\{a_k\}, \{b_k\}, \{c_k\}, \{d_k\}$ такие, что $z_k = a_k - b_k + i(c_k - d_k) \qquad (k = 1, 2, \cdots)$.

7. Всякая последовательность класса BV сходится; привести пример действительной сходящейся последовательности, имеющей неограниченное изменение.

8*. Если последовательность z_n ограничена, то из неё можно извлечь такую подпоследовательность $w_n = z_{k_n} = u_n + iv_n$, что действительные последовательности u_n, v_n монотонны.

9. Пусть $c_1, \ldots, c_p, d_1, \cdots, d_p$ – комплексные числа. Положим

$$D_0 = 0, D_1 = d_1, D_2 = d_1 + d_2, \cdots, D_p = d_1 + d_2 + \ldots + d_p,$$

$$S = c_1 d_1 + c_2 d_2 + \ldots + c_p d_p, \quad L = \max\{|D_k|, k = 1, 2, \cdots, p\}.$$

а) Доказать соотношения

$$S = c_p D_p - \sum_{k=1}^{p-1} D_k (c_{k+1} - c_k), \tag{1}$$

$$|S| \leqslant L \left(|c_p| + \sum_{k=1}^{p-1} |c_{k+1} - c_k| \right); \tag{2}$$

Равенство (1) называют преобразованием Абеля. Оно представляет дискретный аналог интегрирования по частям. Оценка (2) играют важную роль при изучении рядов вида

$$\sum_{k=1}^{\infty} a_k b_k, \tag{3}$$

где $a_k, b_k \in \mathbb{C}\,(k \in \mathbb{N})$. Вместе с рядом (3) удобно рассматривать ряд

$$\sum_{k=1}^{\infty} b_k. \tag{4}$$

10. Если ряд (4) сходится, а последовательность $\{a_k\}$ имеет ограниченное изменение, то и ряд (3) сходится.

11*. Если для всякого сходящегося ряда (4) получаемый из него ряд (3) также сходится, то последовательность $\{a_k\}$ имеет ограниченное изменение.

12. Пусть последовательность $\{a_k\}$ имеет ограниченное изменение и стремится к нулю. Пусть последовательность частичных сумм ряда (4) ограничена. Тогда ряд (3) сходится.

13. Если действительная последовательность $\{a_k\}$ монотонна и стремится к 0, а последовательность частичных сумм ряда (5) ограничена, то ряд (4) сходится.

14. Пусть последовательность положительных чисел a_k убывает и стремится к 0. Тогда знакочередующийся ряд $a_1 - a_2 + \ldots + (-1)^{k-1} a_k + \ldots$ сходится.

15*. Для теории тригонометрических рядов полезно следующее утверждение. Если последовательность $\{a_k\}$ убывает и стремится к 0, то для любого действительного числа α числовой ряд

$$\sum_{k=1}^{\infty} a_k \sin k\alpha$$

сходится.

16*. Пусть $-1 < m < 0$. Тогда биномиальный ряд

$$1 + \sum_{n=1}^{\infty} \frac{m(m-1)\cdots(m-n+1)}{1 \cdot 2 \cdots n} z^n$$

сходится, если и только если $|z| \leqslant 1, z \neq -1$.

17. Логарифмический ряд

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} z^n$$

сходится, если и только если $|z| \leqslant 1, z \neq -1$.

Ссылки

1 *Демидович Б. П.* Сборник задач и упражнений по математическому анализу. М. : Астрель ; АСТ, 2005. 558 с.

УДК 519.72

Е. В. Коновалов
Ярославский государственный университет им. П. Г. Демидова
E-mail: kinnarts@mail.ru

Современные нейросетевые модели в учебном курсе «Нейронные сети и нейрокомпьютеры»

Рассматриваются некоторые современные нейросетевые модели. Обсуждается научно-методическая целесообразность введения их в учебный курс «Нейронные сети и нейрокомпьютеры».

Библиография: 5 названий.

Ключевые слова: искусственные нейронные сети, обобщенные нейронные элементы, модели биологического нейрона, автогенераторы, пороговые нейроны.

В учебной программе факультета ИВТ для студентов предусмотрено проведение нескольких учебных курсов по нейросетевой тематике. Это стало возможным благодаря созданию на факультете В. В. Майоровым научной школы по данной тематике, чрезвычайно популярной в настоящее время. Обучение такой молодой и бурно развивающейся научной области, как нейронные сети, расположенной вдобавок на стыке различных наук, сопряжено с определенными трудностями. Главные из них, по мнению автора настоящей статьи и преподавателя ряда нейросетевых курсов, следующие.

Во-первых, студентам не хватает междисциплинарных знаний, биологических в частности, и умений сопрягать эти биологические знания с математическими идеями и методами, которые необходимы при моделировании искусственных нейронных сетей. Во-вторых, существует определенный дисбаланс между давно и хорошо известными, но чрезвычайно простыми искусственными нейронными сетями (перцептроны, сеть Хопфилда и др.) и гораздо более сложными биологически ориентированными моделями, вызывающими к настоящему моменту наибольший интерес. Простые нейронные сети нашли свое применение в ряде прикладных задач, но использование их в более сложных когнитивных

задачах наталкивается на серьезные проблемы, связанные с принципиальной простотой нейронного элемента — обычного сумматора. Желательно, чтобы обучающиеся постепенно получили представление об обоих типах нейронных моделей, начав с наиболее простых, а затем перейдя и к более сложным. Такова структура и большинства современных учебников по нейронным сетям (например, [1]).

Двухступенчатая образовательная модель предоставляет для этого хорошую возможность. Так, на четвертом курсе предусмотрен предмет «Математические модели нейросетей», содержание которого целесообразно наполнить первоначальным знакомством с нейронными сетями, необходимыми биологическими сведениями, а также разбором наиболее популярных нейронных моделей, таких как однослойный и многослойный перцептроны, сеть Хопфилда и др. Затем для магистрантов второго года обучения предусмотрен курс «Нейронные сети и нейрокомпьютеры». Естественно построить программу этого курса с упором на изучение более сложных биологически ориентированных моделей.

Изучение биологически ориентированных нейронных моделей целесообразно начать с известной модели Ходжкина–Хаксли, подробно описывающей процесс генерации нервного импульса (спайка). В процессе биологических исследований было обнаружено, что транспорт ионов (в соответствии с градиентом концентраций и направлением электрического поля) подчиняется специфическим закономерностям. Для их объяснения А. Ходжкиным и Э. Хаксли была высказана гипотеза о наличии в клеточной мембране специальных ионных каналов, служащих для транспортировки ионов. В процессе исследования мембраны гигантского аксона кальмара А. Ходжкин и Э. Хаксли в 1952 году разработали модель генерации спайка, названную их именами [2].

Для описания процесса генерации потенциала действия ими была предложена система четырех уравнений. Первое из них — электрохимическое уравнение для баланса мембранных токов. Второе и третье уравнения описывают соответственно переменные активации и инактивации натриевых каналов. Наконец, четвертое уравнение отражает динамику переменной активации калиевых каналов.

Используя имеющиеся биологические предпосылки для моделирования, в 1990 году В. В. Майоров предложил модифицированную модель импульсного нейрона [3]. В ее основе также лежит калиево-натриевый цикл. В новой модели учтен тот факт, что ток ионов натрия регулируется двумя видами ворот: открывающимися m-воротами и запаздывающими по отношению к ним закрывающимися h-воротами. На основе предложенной модели В. В. Майоровым также были разработаны специальные модели импульсных нейронов автогенераторного и детекторного типа.

На основе анализа динамики импульсного нейрона Г. В. Шабаршиной была разработана феноменологическая модель нейронного клеточного автомата. Эта модель была предложена в 1994 году в работе [4] и исследована в цикле последующих статей.

Особенности модели заключаются в следующем. Спайк превращен в мгновенный импульс. Во время рефрактерного периода мембранный потенциал не меняется. Также введен экспоненциально падающий порог генерации. Нейронный клеточный автомат является разновидностью интегративно-порогового элемента.

На пути дальнейших упрощений модели импульсного нейрона в 2007 году в работе [5] была предложена новая феноменологическая модель — обобщенный нейронный автомат (ОНА). По мере исследований эта модель без каких-либо содержательных изменений стала называться обобщенным нейронным элементом (ОНЭ). Предлагаемая модель обобщенного нейронного элемента носит универсальный характер: в зависимости от выбранных параметров ОНЭ может вести себя и как нейрон-детектор и как нейрон-автогенератор.

Модель обобщенного нейронного элемента близка к уже рассмотренной импульсной модели нейрона. При этом модель обобщенного нейронного элемента отличается простотой описания и позволяет избежать технических трудностей, связанных с интегрированием систем дифференциальных уравнений с запаздыванием. Кроме того, сети на основе данной модели способны решать более широкий класс задач, чем специализированные сети детекторов или автогенераторов.

Изучение всех этих моделей потребует и биологических, и математических знаний, а также методов аналитических и численных (компьютерных) исследований. Овладение ими позволит подготовить студентов к самостоятельному мышлению и научной деятельности в настоящей области, а также за ее пределами, поскольку на примере этих нейросетевых моделей можно продемонстрировать методологию решения сложных задач и научного поиска в целом.

Ссылки

1. *Хайкин С.* Нейронные сети. М. : Вильямс, 2006. 1104 с.

2. *Hodgkin A. L., Huxely A. F.* A quantitative description of membrane current and its applications to conduction and excitation in nerve // J. Physiol. Vol. 117. L., 1952. P. 500–544.

3. *Майоров В. В., Мышкин И. Ю.* Математическое моделирование нейронов сети на основе уравнений с запаздыванием // Математическое моделирование. 1990. Т. 2, № 11. С. 64–76.

4. *Шабаршина Г. В.* Проведение возбуждения по кольцевой структуре нейронных клеточных автоматов // Моделирование и анализ информационных систем. 1994. № 2. С. 116–121.

5. *Майоров В. В., Коновалов Е. В.* Обобщенный нейронный автомат в задаче распространения волны возбуждения по нейронной сети // Нейрокомпьютеры : Разработка, применение. № 7. М. : Радиотехника, 2007. С. 3–8.

УДК 517.927.4

Е. П. Кубышкин
Ярославский государственный университет им. П. Г. Демидова
E-mail: kubysh@uniyar.ac.ru

Использование методов комплексного анализа в построении решений задачи Дирихле и бигармонической краевой задачи

Изложен подход к построению решений задачи Дирихле и бигармонической краевой задачи, основанный на использовании теории аналитических функций комплексного переменного. Этот подход позволяет с единых позиций посмотреть на задачу построения решений указанных краевых задач.

Библиография: 1 название.

Ключевые слова: задача Дирихле, бигарманическая краевая задача, комплексный анализ.

В плоской ограниченной области Ω с гладкой границей Γ рассматриваются следующие краевые задачи

$$\Delta u = 0,\ u|_\Gamma = f(s), \tag{1}$$

$$\Delta^2 v = 0,\ v|_\Gamma = f_1(s),\ \partial v/\partial n = f_2(s), \tag{2}$$

где $u = u(x,y), v = v(x,y), (x,y) \in \Omega, \Delta u \equiv u_{xx} + u_{yy}$ оператор Лапласа, $\Delta^2 v \equiv v_{xxxx} + 2v_{xxyy} + v_{yyyy}$, параметр $s \in [s_0, s_1]$ характеризует точку границы Γ, n – направление внешней нормали к границе области Ω, функции $f(s), f_1(s), f_2(s) \in C^2[s_0, s_1]$.

Функции, удовлетворяющие уравнению (1), называются гармоническими, уравнению (2) – бигармоническими. Общая теория гарманических функций подробно изложена, например, в [1].

Введем комплексную переменную $z = x+iy,\ (i = \sqrt{-1})$ и перепишем краевые задачи (1),(2) в терминах аналитических функций переменной z. С учетом $x = (z+\bar{z})/2, x = (z-\bar{z})/(2i)$ запишем

$$u = u((z+\bar{z})/2, (z-\bar{z})/(2i)), v = v((z+\bar{z})/2, (z-\bar{z})/(2i)) \tag{3}$$

и будем рассматривать (3) как функции z и $\bar{z}$. Из (3) имеем $u_z = (u_x - iu_y)/2,\ u_{\bar{z}} = (u_x + iu_y)/2$ и

$$\Delta u \equiv u_{xx} + u_{yy} \equiv 4u_{z\bar{z}}, \quad \Delta^2 v \equiv v_{xxxx} + 2v_{xxyy} + v_{yyyy} \equiv 16v_{zz\bar{z}\bar{z}}.$$

Таким образом, уравнения (1) и (2) соответственно примут вид

$$u_{z\bar{z}} = 0, \tag{4}$$

$$v_{zz\bar{z}\bar{z}} = 0. \tag{5}$$

Уравнения (4) и (5) легко интегрируются. Интегрируя (4) по $\bar{z}$, получим уравнение $u_z = \varphi_1(z)$, где $\varphi_1(z)$ – произвольная функция z. Интегрируя теперь это уравнение по z, будем иметь

$$u(z, \bar{z}) = \int_{z_0}^{z} \varphi_1(z_1)dz_1 + \varphi_2(\bar{z}), \tag{6}$$

где $\varphi_2(\bar{z})$ – произвольная функция $\bar{z}$. Так как $u(z, \bar{z})$ – гармоническая функция, являющаяся вещественной частью функии комплексного переменного, то, согласно (6), с необходимостью имеем выражение

$$u(z, \bar{z}) = (\varphi(z) + \overline{\varphi(z)})/2, \tag{7}$$

где $\varphi(z)$ – произвольная аналитическая в области Ω функция z.

Интегрируя аналогичным образом уравнение (5), получим общее решение уравнения (5) в виде

$$v(z, \bar{z}) = (\varphi_1(z) + \overline{\varphi_1(z)} + \bar{z}\varphi_2(z) + z\overline{\varphi_2(z)})/2, \tag{8}$$

где $\varphi_1(z)$ и $\varphi_2(z)$ – произвольные аналитические в области Ω функции z.

Граничное условие в (1) для функции (7) примет вид

$$(\varphi(z) + \overline{\varphi(z)})|_\Gamma = 2f(s). \tag{9}$$

Задача (1) свелась, таким образом, к отысканию аналитической в области Ω функции $\varphi(z)$, действительная часть которой обращается на границе Γ в заданную функцию $f(s)$.

Выразим граничные условия в (2) через функции $\varphi_1(z)$ и $\varphi_2(z)$. Заметим, что $v_x = v_z + v_{\bar{z}},\ v_y = (v_z - v_{\bar{z}})i$. Отсюда

$$\partial v/\partial n = v_x \cos(n \wedge x) + v_y \cos(n \wedge y) = v_z\sigma(z) + v_{\bar{z}}\overline{\sigma(z)},$$

где $\sigma(z) = \cos(n \wedge x) + i\cos(n \wedge y)$ – известная на Γ функция. С учетом этого граничные условия в (2) можно записать в следующем виде

$$(\varphi_1(z) + \overline{\varphi_1(z)} + \bar{z}\varphi_2(z) + z\overline{\varphi_2(z)})|_\Gamma = 2f_1(s), \tag{10}$$

$$[(\varphi_1{}'(z) + \overline{\varphi_2(z)} + \bar{z}\varphi_2{}'(z))\sigma(z) + \\ + (\overline{\varphi_1{}'(z)} + \varphi_2(z) + z\overline{\varphi_2{}'(z)})\overline{\sigma(z)}]|_\Gamma = 2f_2(s). \tag{11}$$

Таким образом, задача (2) свелась к отысканию двух аналитических в области Ω функций, удовлетворяющих на границе Γ условиям (7), (8).

В общем случае задача нахождения функции $\varphi(z)$, удовлетворяющей условию (9), и функций $\varphi_1(z)$, $\varphi_2(z)$, удовлетворяющих условиям (10), (11), является непростой задачей. Рассмотрим эту задачу в предположени, что область Ω является кругом K_R радиуса R с центром в нуле. Точку на границе K_R обозначим $\zeta = R\exp(i\theta)$, $\theta \in [0, 2\pi)$. Имеем $f(s) = f(R\theta) = f'(\theta)$ (штрих в дальнейшем опустим). В результате равенство (9) может быть записано в виде

$$\varphi(\zeta) + \overline{\varphi(\zeta)} = 2f(\theta), \theta = -i\ln(\zeta/R)\ (\ln(z) = \ln|z| + i\arg(z)). \qquad (12)$$

Умножим равенство (12) на $d\zeta/[2\pi i(\zeta - z)]$ и проинтегрируем по Γ. В результате имеем

$$\frac{1}{2\pi i}\int_\Gamma \frac{\varphi(\zeta)d\zeta}{\zeta - z} + \frac{1}{2\pi i}\int_\Gamma \frac{\overline{\varphi}(\zeta)d\zeta}{\zeta - z} = \frac{1}{\pi i}\int_\Gamma \frac{f(\theta)d\zeta}{\zeta - z}. \qquad (13)$$

Первый интеграл в (13) по теореме Коши определяет функцию $\varphi(z)$. Покажем, что второй интеграл равен $\overline{\varphi(0)}$. Представим $\bar{\zeta}$ с учетом равенства $\bar{\zeta} = R^2/\zeta$ ($\zeta = R\exp(i\theta)$) на границе Γ

$$\varphi(\zeta) = \sum_{n=0}^{\infty} a_n\zeta^n, \overline{\varphi(\zeta)} = \sum_{n=0}^{\infty} \bar{a}_n\bar{\zeta}^n = \sum_{n=0}^{\infty} \bar{a}_n R^{2n}/\zeta^n.$$

Отсюда с учетом равенства $\bar{a}_0 = \overline{\varphi}(0)$ получим

$$\frac{1}{2\pi i}\int_\Gamma \sum_{n=0}^{\infty} \frac{\bar{a}_n R^{2n}d\zeta}{\zeta^n(\zeta - z)} = \overline{\varphi}(0)(\frac{1}{2\pi i}\int_\Gamma \frac{d\zeta}{\zeta - z} = 1, \frac{1}{2\pi i}\int_\Gamma \frac{d\zeta}{\zeta^n(\zeta - z)} = 0)$$

(n>0). В связи с этим и согласно (13)

$$\varphi(0) + \overline{\varphi(0)} = \frac{1}{\pi i}\int_\Gamma \frac{f(\theta)d\zeta}{\zeta}.$$

В итоге имеем

$$u(z, \bar{z}) = \frac{1}{2\pi i}\int_\Gamma \frac{f(\theta)d\zeta}{\zeta - z} - \frac{1}{2\pi i}\int_\Gamma \frac{f(\theta)d\bar{\zeta}}{\bar{\zeta} - \bar{z}} - \frac{1}{2\pi i}\int_\Gamma \frac{f(\theta)d\zeta}{\zeta}. \qquad (14)$$

Введем в K_R полярные координаты $z = \rho\exp(i\phi)$ и положим $u(\rho, \phi) = u(\rho\exp(i\phi), \rho\exp(-i\phi))$. При этом будем иметь $d\zeta = iR\exp(i\theta)d\theta = i\zeta d\theta$. В результате с учетом равенства

$$\frac{1}{2\pi}\left(\frac{R\exp(i\theta)}{R\exp(i\theta) - \rho\exp(i\phi)} + \frac{R\exp(-i\theta)}{R\exp(-i\theta) - \rho\exp(-i\phi)} - 1\right) =$$
$$= \frac{1}{2\pi}\frac{R^2 - \rho^2}{R^2 + \rho^2 - 2R\rho\cos(\theta - \phi)} \equiv K(\rho, \phi, R, \theta)$$

выражение (14) примет вид

$$u(\rho,\phi)=\int_0^{2\pi}K(\rho,\phi,R,\theta)f(\theta)d\theta, \tag{15}$$

т. е. получили известную формулу Пуассона. Функция $K(\rho,\phi,R,\theta)$ при этом носит название ядра Пуассона.

Рассмотрим теперь в круге K_R бигармоническую краевую задачу. Представим функции, входяшие в (8), в виде

$$\varphi_1(z)=\sum_{n=0}^{\infty}a_nz^n,\varphi_2(z)=\sum_{n=0}^{\infty}b_nz^n.$$

Заметим, что

$$z\overline{\varphi_2(z)}+\bar z\varphi_2(z)=\bar b_0z+b_0\bar z+z\bar z\sum_{n=1}^{\infty}(\bar b_nz^{n-1}+b_n\bar z^{n-1}).$$

Это дает возможность представление (8) записать в следующем виде

$$v(z,\bar z)=[\varphi_1^*(z)+\overline{\varphi_1^*(z)}+z\bar z(\varphi_2^*(z)+\overline{\varphi_2^*(z)})]/2, \tag{16}$$

где $\varphi_1^*(z)$ и $\varphi_2^*(z)$ – произвольные аналитические в K_R функции z. Перейдя теперь в K_R к полярным координатам, выражение (16) для $v(\rho,\phi)=v(\rho\exp(i\phi),\rho\exp(-i\phi))$ запишем следующим образом

$$v(\rho,\phi)=v_1(\rho,\phi)+(\rho^2-R^2)v_2(\rho,\phi), \tag{17}$$

где $v_1(\rho,\phi),v_2(\rho,\phi)$ – произвольные гармонические функции. Представим $f_j(s)=f_j(R\phi)=f_j'(\phi),j=1,2$ (штрих в дальнейшем опустим). Из (17) и первого граничного условия (2) имеем $v(R,\phi)=v_1(R,\phi)=f_1(\phi)$. Таким образом, согласно (16),

$$v_1(\rho,\phi)=\int_0^{2\pi}K(\rho,\phi,R,\theta)f_1(\theta)d\theta.$$

Выражение (17) и второе граничное условие в (2) дают

$$v_2(R,\phi)=(2R)^{-1}(f_2(\phi)-\partial v_1(\rho,\phi)/\partial\rho|_{\rho=R}).$$

В итоге решение задачи (2) для круга K_R дается выражением

$$v(\rho,\phi)=\int_0^{2\pi}K(\rho,\phi,R,\theta)[f_1(\theta)+$$
$$+(2R)^{-1}(R^2-\rho^2)(\int_0^{2\pi}K_\rho(R,\theta,R,\theta_1)f_1(\theta_1)d\theta_1-f_2(\theta))]d\theta. \tag{18}$$

В заключение отметим следующее. Если произвольная области Ω с гладкой границей может быть конформно отображена в круг K_R, то решение задач (1), (2) для такой области стороится суперпозицией конформного отображения и выражений (15), (18).

Ссылки

1. *Смирнов М. М.* Дифференциальные уравнения в частных производных второго порядка. М. : Наука, 1964.

УДК 001.8

А. Н. Куликов, Д. А. Куликов
Ярославский государственный университет им. П. Г. Демидова
E-mail: kulikov_d_a@mail.ru
E-mail: kulikov@uniyar.ac.ru

Понятийное мышление и математическое образование

Обсуждается проблема снижения уровня подготовки студентов. Одной из основных причин следует признать не само введение ЕГЭ в средней школе, а отсутствие у школьников и студентов навыков понятийного мышления (понятие, введенное известным психологом Львом Выготским).

Библиография: 2 названия.

Ключевые слова: математическое образование, логическое и понятийное мышления, системное образование.

Снижение уровня подготовки студентов, поступивших на математический, ИВТ, физический факультеты, в целом общепризнано. Этоимеет место не только в Ярославском государственном университете, но и в других университетах России. В равной мере это относится к университетам других стран. Хотя, конечно, возможны исключения и контрпримеры, но в большинстве университетов это так. Данная проблема имеет много различных проявлений. Например, она сказывается при наборе студентов на специальности, связанные с точными науками: математикой, физикой, информатикой. Кстати, особенно плохо дело обстоит с формированием набора на специальности физического профиля, так как школьники отказываются сдавать ЕГЭ по физике. В свою очередь, стойкое игнорирование тестов по физике связано с нововведениями на уроках физики, где минимизированы задания по решению задач. «Очень передовые» школы перешли на систему, где школьники делают доклады и "презентации". Достаточно сходная картина при обучении математики и информатики. По-видимому, причина здесь не только в введении ЕГЭ, а напротив, введение ЕГЭ – это следствие изменения системы образования в школе и вузовском образовании. Недаром в университетах с удовольствием внедряют интернет-тестирование,

и уже идут разговоры о внедрении государственной аттестации в форме сходной с ЕГЭ в средней школе.

Преподаватели в университетах уже не один год отмечают понижение уровня математической подготовки и в университетах Европы и США. Приведем один, пожалуй, самый известный пример из статей В. И. Арнольда [1]. В одном из университетов Франции студенты четвертого курса математического факультета испытывали затруднения при ответе на такой вопрос: «Что больше 1 или 4/7». Об уровне подготовки школьников в области математики совсем нечего говорить. В Швеции на общенациональной контрольной по математике только треть школьников справились со следующей задачей: масштаб карты 1:500000, расстояние между городами 6 см. Сколько километров разделяет города в действительности. Заметим, что Швеция в первой половине XX века была одной из самых передовых стран в мире в сфере образования и поэтому одной из первых перешла на «сверхсовременные образовательные технологии» родом из США, а ее бывшая колония – Финляндия, традиционно отстала и до сих пор пользуется старыми образовательными технологиями. В настоящее время в Европе признали уровень подготовки финских школьников лучшим на континенте.

Какие же элементы выпали из системы образования в школе и, к сожалению, начинают выпадать в университетских методиках? Понятно, что не один, и некоторые следует отметить. Процесс образования перестал обращать внимание на такой элемент интеллектуального развития, как «понятийное мышление». Этот термин был введен в работах советского психолога Льва Выготского [2]. В этих работах отмечалось, что понятийное мышление определяется через три важных компонента:

- умение выделять суть явления, объекта, понятия, метода;

- умение видеть причину и прогнозировать последствия;

- умение систематизировать информацию и строить целостную картину ситуации.

Согласно исследованиям психологов, лишь 20 % (общемировая статистика) людей обладают полноценным понятийным мышлением. Как правило, это те, кто изучал естественные, точные и технические науки, научился операциям выделения существенных признаков и установления причинно-следственных связей. В математике часть этих навыков объединяется понятием – умение мыслить логически. При этом подчеркнем, что умение мыслить логически – это важный, но не единственный аспект понятийного мышления. Сюда следует отнести, например, умение обобщать, переходить от частного к общему.

Приведем два примера. Обучение решению задач на основе метода Фурье фактически может быть рассказано один раз. Допустим, для уравнений параболического типа. Для аналогичных задач при рассмотрении уравнений эллиптического и гиперболического типов все в доста-

точной мере повторяется, но не на уровне прямолинейных аналогий, а слегка опосредованных, а также завуалированных. Тем не менее последние годы при освоении этого метода студенты испытывают все большие затруднения и приходится достаточно детально повторять соответствующую методику минимум три раза. А попытки объяснения, основанные на аналогиях, как правило, абсолютно бесполезны.

Аналогичная ситуация возникает при обучении программированию. Последние 10 лет студенты с явным одобрением относятся к идее изучить еще один язык высокого уровня. Сначала допустим Pascal, затем C++, *C♯* и т. д. Как правило, их интузиазм базируется на желании освоить хотя бы один язык, после того как с предшествующим одним уже не очень получилось. Здесь явно прослеживается неумение находить содержательные, неслучайные аналогии.

Изменить что-либо в системе образования в средней школе вряд ли в силах преподавателей высшей школы, а тем более преподавателей ЯрГУ, но вместе с тем ситуация требует определенной корректировки. Лекционные курсы базовых дисциплин, по нашему мнению, следует адаптировать к той ситуации, которая была описана выше. Последнее замечание означает, что больший объем времени должен быть направлен на развитие понятийного мышления. Как уже отмечалось, это в первую очередь включает такой элемент, как обучение приемам и методам логического мышления. При этом имеются в виду и начальные приемы и методы, которые, как принято считать, должны быть освоены на стадии обучения в средней школе.

Ссылки

1. *Арнольд В. И.* Путешествие в хаосе // Наука и жизнь. 1999. № 12. С. 2–10.

2. *Выготский Л. С.* Мышление и речь. М. : Наука, 1956.

УДК 378.16, 004.422.8

Н. С. Лагутина
Ярославский государственный университет им. П. Г. Демидова
E-mail: lagutinans@rambler.ru

Разработка компьютерных обучающих программ для развития иноязычной коммуникативной и межкультурной компетенций студентов*

В работе рассматривается необходимость формирования профессиональных компетенций студентов IT-специальностей в области знания иностранных языков, главным образом английского. Для решения проблемы описывается проект по созданию электронных образовательных ресурсов, разрабатываемый совместно с кафедрой иностранных языков.

Библиография: 1 название.

Ключевые слова: электронный образовательный ресурс, английский язык.

Современное общество характеризуется бурным развитием технологий, особенно информационных. Выпускники вуза по направлениям, связанным с информационными технологиями, например «Прикладная математика и информатика» и «Фундаментальная информатика и информационные технологии», должны обладать общекультурными и профессиональными компетенциями в области владения иностранными языками, в первую очередь, английским языком.

Английский язык (см., например, [1]) – язык века информации. Компьютеры разговаривают друг с другом на английском. Более 80 % всей информации в более чем 200 млн компьютерах по всему свету хранится на английском языке; 85 % всех международных телефонных разговоров совершаются на английском языке; так же как и три четверти мировой почты, телексов и телеграмм. Инструкции к компьютерным программам и сами программы часто бывают только на английском языке. Когда-то языком

*Работа выполнена при поддержке проекта № 549 в рамках базовой части государственного задания на НИР ЯрГУ.

науки были латинский, немецкий, сегодня 85 % всех научных работ публикуются сначала на английском языке. Более половины мировых технических и научных периодических изданий выходят на английском языке, который также является языком медицины, электроники и космической технологии.

Без освоения английского языка заниматься профессионально техническими науками невозможно; если его не знать, то спектр материалов для изучения сужается во много раз. Кроме того, английский нужно знать, чтобы понимать синтаксис языков программирования, а также чтобы успешно общаться в мировом информационно-технологическом сообществе. При этом мало понимать специализированные статьи и документацию, так называемый «технический английский язык», а необходимо знать язык на разговорном уровне.

Таким образом, можно выделить ряд умений и навыков, необходимых программистам в рамках владения английским языком:

1. Локализовывать (переводить) программное обеспечение;
2. Переводить тексты по информационным технологиям на русский и английский языки и не только;
3. Пользоваться на продвинутом уровне информационными и телекоммуникационными технологиями, дистанционными образовательными технологиями и прикладными программами;
4. Извлекать информацию из любого источника на английском языке, структурировать ее и трансформировать с учетом цели коммуникации;
5. Профессионально создавать и оформлять контент на нескольких языках, создавать и оптимизировать содержание сайтов;
6. Общаться на любые, в том числе профессиональные темы.

Получить все эти знания и умения только во время занятий по обычному для студентов курсу иностранного языка невозможно. Одним из решений проблемы может быть интеграция усилий преподавателей филологии, иностранных языков и программирования в рамках совместных проектов. Таким проектом является исследование и разработка электронных учебных ресурсов по английскому и другим языкам, а также разработка программного обеспечения, связанного с общением людей, например в области туризма. Во время работы над проектом ставятся следующие задачи:

1. Разработка инновационных электронных учебных пособий, способствующих развитию коммуникативных умений студентов;

2. Исследование влияния использования современных информационных технологий в процессе обучения и формирования иноязычных коммуникативных компетенций;

3. Создание базы данных учебных материалов и электронных образовательных ресурсов, разработка компьютерных обучающих программ по развитию иноязычной коммуникативной и межкультурной компетенций;

4. Исследование методик обучения иностранному языку в вузе с использованием информационных технологий, выбор материалов для разработки электронных образовательных ресурсов;

5. Разработка электронной базы данных учебных материалов, разработка прототипов программных приложений для обучения иностранному языку;

6. Тестирование и анализ результатов использования электронных образовательных ресурсов.

Современные информационные и коммуникационные технологии позволяют решить ряд задач, направленных на активизацию учебной деятельности и развитие культуры самостоятельной работы студентов. Применение инноваций в процессе обучения иностранному языку способствует повышению мотивации обучающихся, реализации дифференцированного подхода к обучению, его индивидуализации, повышению эффективности самостоятельной работы студентов и контроля знаний, умений, навыков по предмету. С другой стороны, создание соответствующих программных приложений заставляет студентов-программистов повышать свой уровень знания английского языка и следовать международным стандартам разработки компьютерных программ, стимулирует учащихся к дальнейшему самостоятельному изучению иностранного языка.

Ссылки

1. *Crystal D.* English as a global language. Cambridge : Cambridge University Press, 2003.

УДК 519.67

А. Н. Максименко, А. Ю. Ухалов
Ярославский государственный университет им. П. Г. Демидова
E-mail: maksimenko_a_n@mail.ru
E-mail: alex@ukhalov.com

Использование технологий 3D-печати в учебном процессе

В этой краткой заметке мы хотим поделиться опытом работы с 3D-принтерами, имеющимися в лаборатории "Дискретная и вычислительная геометрия", и кратко описать возможности их использования в образовательном процессе.

Библиография: 2 названия.

Ключевые слова: 3D-принтер, 3D-моделирование, подготовка моделей, представление поверхностей, CAD.

3D-печать

Первые серийные образцы 3D-принтеров были выпущены фирмой 3D Systems в конце 1980-х гг. Они были основаны на технологии стереолитографии. К концу прошлого века в связи с возросшим к этой сфере интересом появился целый ряд новых технологий 3D-печати (см., например, http://3dcorp.ru/story.html). В настоящее время 3D-принтеры активно используются в индустрии: начиная от печати прототипов различных объектов и заканчивая печатью готовых изделий. Еще не охваченными в полной мере, но весьма перспективными областями использования 3D-печати являются образование, дизайн, производство сувенирной продукции, и особенно медицинская сфера (начиная от изготовления имплантантов и заканчивая моделированием человеческих органов для репетиции перед проведением сложных операций). Для более полного представления возможностей этой технологии предлагаем заглянуть на портал www.3dindustry.ru.

Постепенно устройства для 3D-печати становятся доступными для рядовых покупателей. Так, самые дешевые, не требующие специальной

настройки 3D-принтеры от ветерана этой индустрии 3D Systems продаются за 1300$ (см. http://cubify.com/cube/). Этот процесс напоминает вхождение в нашу жизнь персональных компьютеров, или, что значительно ближе, — обычных 2D-принтеров. Тем не менее процесс вхождения 3D-принтеров в "массы" существенно осложняется тем, что подготовка моделей для печати требует от пользователя навыков работы с "неплоскими" объектами, а обычные плоские мониторы персональных компьютеров совсем не облегчают этот процесс.

В 2012 году в рамках создания в ЯрГУ международной научно-исследовательской лаборатории "Дискретная и вычислительная геометрия" (лаборатория ДВГ) под руководством Херберта Эдельсбруннера был куплен современный профессиональный 3D-принтер ZPrinter 450. От других 3D-принтеров того же уровня он отличается высоким качеством печати, относительно низкой стоимостью расходных материалов и возможностью полноцветной печати. В конце 2013 года в лаборатории появились еще два, но уже относительно дешевых настольных 3D-принтера, использующих принципиально иную технологию печати и, как показал опыт, более капризных в работе. За этот небольшой период времени сотрудники лаборатории приобрели достаточно серьезный опыт работы с 3D-принтерами, подготовили для печати и напечатали большое количество различных объектов.

Настоящая работа преследует две цели: знакомство преподавателей ЯрГУ с возможностями использования 3D-печати в образовательном процессе и приглашение к использованию этих возможностей.

Способы подготовки моделей для печати

Для подготовки моделей, как правило, используются специализированные программы. В настоящий момент существует очень большое количество 3D-редакторов разной сложности и назначения. Одни из них бесплатны, другие могут стоить многие тысячи долларов. Опыт авторов, однако, показывает, что пригодность того или иного программного продукта для подготовки моделей требуется проверять экспериментально для каждого из используемых 3D-принтеров. Тот факт, что модель хорошо выглядит на экране компьютера, еще не гарантирует того, что она будет корректно прочитана программой печати принтера.

Далее в данной статье описываются некоторые из способов создания моделей, опробованные в лаборатории ДВГ в 2012–2013 годах. Приведенные сведения являются далеко не исчерпывающими, но достаточны для начала самостоятельной работы по подготовке моделей.

Подготовка моделей на низком уровне

При подготовке моделей для печати далеко не всегда требуется разбираться в подробностях внутреннего представления моделей. Однако понимание основных принципов и знание простейших форматов представления данных очень полезны. Это позволяет проанализировать причины возникающих проблем и при необходимости создать модель “своими руками”, воспользовавшись текстовым редактором типа Блокнота или написав свою простую программу на одном из языков программирования.

Программное обеспечение 3D-принтеров способно воспринимать много различных форматов файлов, описывающих 3D-объекты. Однако как правило, при подготовке модели к печати программа, управляющая печатью, производит триангуляцию поверхности, ограничивающей объект. Многие форматы данных, собственно, и содержат описание такой триангуляции.

Простейшим форматом для хранения триангуляции является формат STL. Файлы STL бывают текстовыми (ASCII STL) и двоичными. Двоичные файлы содержат ту же информацию, что и текстовые, но позволяют значительно уменьшить объем хранимых данных.

Приведем краткое описание формата ASCII STL файлов. Более детальную информацию можно легко найти в сети Internet (см., например, http://ru.wikipedia.org/wiki/STL_(формат_файла) или в статьях [1] и [2].

Файл STL имеет следующую структуру:

```
solid object_name
...
endsolid object_name
```

Между открывающей и закрывающей командами приводится список треугольников (фасет). Описание одного треугольника имеет вид:

```
facet normal nx ny nz
outer loop
vertex x1 y1 z1
vertex x2 y2 z2
vertex x3 y3 z3
endloop
endfacet
```

Здесь $(x1, y1, z1)$, $(x2, y2, z2)$, $(x3, y3, z3)$ – координаты вершин треугольника, (nx, ny, nz) – нормаль треугольника (вектор единичной длины, направленный наружу).

При описании треугольников следует обратить внимание на два важных требования:

- Порядок перечисления вершин должен соответвовать ориентации треугольника (вершины должны обходиться в направлении против часовой стрелки при взгляде со стороны нормали).
- Если два треугольника имеют общие точки, то это либо одна вершина, либо целая сторона. Другие варианты недопустимы.

Стандартный формат файлов STL не предусматривает хранения данных о цвете. Если 3D-принтер позволяет печатать в цвете, объект можно раскрасить (или наложить на поверхность текстуру) непосредственно перед печатью, используя программное обеспечение принтера.

Приведем простой пример объекта в формате ASCII STL – четырехгранной пирамиды.

```
solid my_surface
facet normal  0   0  -1
  outer loop
    vertex    0   0   0
    vertex    1   1   0
    vertex    1   0   0
  endloop
endfacet
facet normal  0   0  -1
  outer loop
    vertex    0   0   0
    vertex    0   1   0
    vertex    1   1   0
  endloop
endfacet
facet normal  0  -1   0
  outer loop
    vertex    0   0   0
    vertex    1   0   0
    vertex    0   0   1
  endloop
endfacet
facet normal -1   0   0
  outer loop
    vertex    0   0   0
    vertex    0   0   1
    vertex    0   1   0
  endloop
endfacet
facet normal 0.7071 0 0.7071
  outer loop
```

```
    vertex     0    0    1
    vertex     1    0    0
    vertex     1    1    0
  endloop
endfacet
facet normal 0 0.7071 0.7071
  outer loop
    vertex     0    0    1
    vertex     1    1    0
    vertex     0    1    0
  endloop
endfacet
endsolid
```

Вид построенного объекта показан на рис. 1.

Для просмотра полученного файла можно скопировать приведенный код в текстовый файл, сохранить файл с расширением “.stl” и открыть файл в любом 3D редакторе, поддерживающем соответствующий формат файлов. Доступно большое количество бесплатных программ, предназначенных для просмотра файлов формата STL (STLViewer, STLView и т. д.).

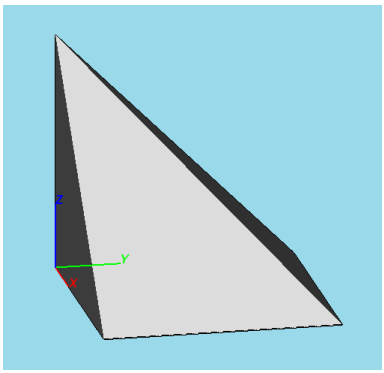

Рис. 1. Модель “пирамида”

Простота этого способа создания моделей открывает широкие возможности для собственных экспериментов. Собственный генератор STL файлов может быть написан студентом в рамках учебного задания. Более сложные программы можно поручить разработать в качестве курсовой или дипломной работы. На рис. 2 показан набор моделей, созданных с помощью собственных программ, написанных сотрудниками лаборатории ДВГ.

Рис. 2. Модели, подготовленные в лаборатории ДВГ с помощью собственных программ и напечатанные на 3D-принтере ZPrinter 450. Поверхности, ограничивающие тела, состоят из большого количества треугольников, выведенных в STL файл

Подготовка моделей с помощью стандартных программ

В лаборатории ДВГ нами были опробованы многие программные продукты. Наиболее активно используются программы Autodesk 3ds Max, Geomagic Studio и Blender. В настоящей заметке невозможно даже кратко описать процесс создания моделей с помощью этих программных продуктов. Документация по их использованию занимает тысячи страниц. Из того, что нам удалось попробовать, мы бы порекомендовали программу Blender, которая, несмотря на то что она бесплатна, обладает большими возможностями. В частности, имеются надстройки Blender, позволяющие задавать поверхности и кривые уравнениями.

Опишем еще один способ создания 3D-моделей, удобный в первую очередь для математиков. Многие пользователи, возможно, не знают, что в популярной программе Wolfram Mathematica предусмотрены возможности экспорта моделей. Соответствующая директива имеет формат

```
Export[file, expr, format]
```

Здесь file – имя файла, expr – выражение, описывающее объект для экспорта, format – формат экспорта. Последний параметр может быть опущен. В этом случае формат для экспорта определяется расширением имени файла.

Приведем пример экспорта модели, напечатанной в лаборатории ДВГ в период знакомства с возможностями 3D-принтера. Функция, описывающая два зацепленных тора, взята из документации программы Wolfram Mathematica.

```
rings = ParametricPlot3D[
{{4+(3+Cos[v]) Sin[u],4+(3+Cos[v]) Cos[u],4+Sin[v]},
{8+(3+Cos[v]) Cos[u],3+Sin[v],4+(3+Cos[v]) Sin[u]}},
{u,0,2 Pi},{v,0,2 Pi}];

Export["ringsfile.stl", rings]
```

Модель, полученная в результате выполнения этого кода, сохраненная в файле ringsfile.stl, показана на рис. 3.

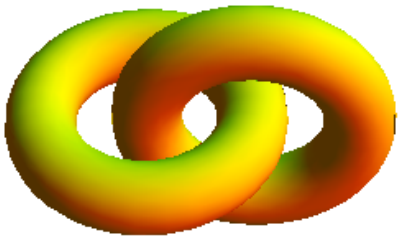

Рис. 3. Модель, подготовленная с помощью программы Wolfram Mathematica

Опыт авторов показывает, что задания по подготовке моделей для 3D-печати вызывают большой интерес у студентов. Выполнение таких заданий способствует активизации знаний по аналитической геометрии, алгебре и программированию.

Ссылки

1. *Al Dean.* STL – формат для быстрого прототипирования. Часть I. Вывод в формате STL // CAD/CAM/CAE Observer. 2005. № 5 (23). С. 64 – 69.

2. *Al Dean.* STL – формат для быстрого прототипирования. Часть II. Реальный опыт вывода STL-файлов // CAD/CAM/CAE Observer. 2005. № 6 (24). С. 65 – 69.

УДК 517.2

Н. Л. Майорова, Г. В. Шабаршина
Ярославский государственный университет им. П. Г. Демидова
E-mail: mnlv@yandex.ru
E-mail: shegeve@yandex.ru

Изучение и повторение отдельных тем дисциплины «Математический анализ»

В заметке рассмотрен вопрос об использовании методического приема регулярного возвращения к пройденному материалу, но на более высоком уровне, на примере решения задач курса математического анализа.

Библиография: 1 название.

Ключевые слова: асимптота графика функции, формула Тейлора, преемственность в обучении.

Раздел математического анализа «Дифференциальное исчисление» начинают изучать еще в десятом классе средней школы. Учат школьников вычислять производные простейших сложных функций, составлять уравнения касательных к кривым, находить промежутки монотонности функций и их экстремумы. Затем в выпускном классе школьники повторяют пройденное в рамках применения его при сдаче ЕГЭ, то есть решают пару типов задач: по графику функции или ее производной ответить на несложные вопросы касательно количества точек экстремума, промежутков монотонности определенного характера, величины углового коэффициента касательной и т. п. Отрабатывается также задача о нахождении наибольшего или наименьшего значения функции на заданном отрезке. Однако этот материал является весьма сложным для восприятия учащихся. В большинстве случаев они применяют его лишь на уровне алгоритмических действий, по аналогии с решенными задачами. Стоит лишь слегка изменить условие, переформулировать его, и задача останется нерешенной. Вообще говоря, в рамках средней школы можно в общеобразовательных классах не изучать дифференциальное и интегральное исчисление, так как профессионально ориентированные учащиеся будут подробно изучать этот раздел в высшей школе, а в средней оставить больше времени на закрепление других тем. Как уже

было сказано, материал этот сложный и без многих теоретических обоснований непонятен большинству подростков, тем более многие учителя в сложившейся в стране ситуации не приучают их читать учебники, понимать доказательства теорем, выводить нужные формулы. Все это весьма усложняет работу педагога высшей школы, когда к нему приходят вчерашние школьники, которым нужно практически моментально изменить привычные для них методы учебы.

И вот начинается первый семестр, изучение математического анализа и опять дифференциальное исчисление. Но, естественно, вначале надо изучить теорию пределов, замечательные пределы и следствия из них, чтобы понять определение производной, вывод производных элементарных функций, связь знака производной с характером монотонности функции и многое другое. Функции рассматриваются достаточно сложные по сравнению со школой: рациональные, алгебраические, показательные, логарифмические, тригонометрические и обратные тригонометрические функции, а также всевозможные функции, которые получаются из перечисленных с помощью арифметических операций и образования сложной функции. Один из этапов исследования и построения графика — нахождение асимптот. В школе вообще редко занимаются построением графиков функций, кроме, быть может, парабол и синусоид. Студент плохо осознает, где находятся точки графика, сталкиваясь при исследовании с ситуацией, например,

$$\lim_{x\to x_0+0} f(x) = +\infty, \lim_{x\to x_0-0} f(x) = +\infty,$$
$$\lim_{x\to x_0+0} f(x) = -\infty, \lim_{x\to x_0-0} f(x) = -\infty.$$

На практических занятиях преподаватель учит находить асимптоты графиков функций согласно выведенным на лекциях формулам, вычислять односторонние пределы. Первокурсникам сложно все это постигать. Поэтому даже если асимптота найдена в виде формулы, ее взаимодействие с графиком надо долго отрабатывать с большинством студентов. Аналогичные трудности возникают при нахождении и изображении экстремумов функций и точек перегиба. Все требует кропотливых размышлений, повторений, самостоятельной отработки при выполнении домашних заданий. Многие же наши учащиеся на первом курсе теряются от обилия преподаваемых им предметов, их сложности, непривычной подачи материала, тем более что у многих из них на самом деле очень слабое знание даже школьного курса математики и полное нежелание работать дома.

Но предмет изучать надо, одна тема сменяет другую. И обращение к пройденному материалу по мере изучения нового полезно и необходимо, чтобы у студента сложилась полная картина пройденного. Это можно делать по мере изучения курса математического анализа, а можно и потом посмотреть на задачи, когда студент доучился до пятого курса. Ни

для кого не является секретом, что к этому времени будущие выпускники забывают все то, что изучали на младших курсах. С декабря по февраль для них проводятся спецкурсы на математическом факультете или читается цикл лекций перед ГОСами на факультете ИВТ. Здесь идет повторение основ математического анализа, дифференциальных уравнений, алгебры и некоторых других предметов. В целом студенты понимают, что предлагаемые задачи ими ранее решались, но тонкости решения уже забыты (к сожалению, для некоторых учащихся и не только «тонкости»). Например, в 2013/14 учебном году на проверочной контрольной работе по математического анализу на матфаке в первый день занятий из 16 баллов один студент получил 11, 75 баллов, двое — 6,5 баллов и 6 баллов, еще двое — 5,25 и 4, 75 баллов, остальные — не более 3 баллов. На последнем занятии из 32 студентов на аналогичной контрольной работе после аккуратного повторения материала только семеро более-менее правильно построили график функции (на основе полного исследования) $y = \frac{x^2+1}{x}$.

С другой стороны, при повторении материала с четвертым и пятым курсами можно опираться на все факты, определения, теоремы данной дисциплины и многих смежных и надеяться на понимание большей части слушателей. Тем более что близятся государственные экзамены и студент становится очень заинтересованным. Поскольку в программе экзамена есть вопросы о применении дифференциального исчисления к построению графиков функций и о степенных рядах, то наступил момент их совместить и продемонстрировать комиссии хорошее знание предмета.

Ниже рассматривается задача нахождения асимптот графика функции. Мы решаем ее в первом семестре в теме исследования функций и построения их графиков (обычно используется задачник [1]). Затем, когда наступает время изучить вопрос о возможности представления функции многочленом Тейлора, можно еще раз вернуться к пройденному материалу. А именно, повторив определение горизонтальных и наклонных асимптот, показать другую возможность найти асимптоту.

Приведем пример функции

$$y = \ln(e^{2x} - 2e^x + 2),$$

для которой формулы нахождения наклонной асимптоты, рассматриваемые на первом курсе, трудно применимы. Проведем необходимые выкладки:

$$y = \ln e^{2x}(1 - 2e^{-x} + 2e^{-2x}), y = \ln e^{2x} + \ln(1 - 2e^{-x} + 2e^{-2x}),$$
$$y = 2x + \ln(1 - 2e^{-x} + 2e^{-2x}).$$

Далее, воспользовавшись разложением функции

$$\ln(1+t) = t - \frac{t^2}{2} + \cdots + (-1)^{n-1}\frac{t^n}{n} + o(t^n), t \to 0,$$

получим

$$y = 2x + (-2e^{-x} + 2e^{-2x}) + o(-2e^{-x} + 2e^{-2x}).$$

Отсюда определяется поведение функции при $x \to +\infty$ и уравнение правой наклонной асимптоты $y = 2x$. При $x \to -\infty$ существует другая асимптота, горизонтальная $y = \ln 2$. Далее можно определить, что функция имеет единственный гладкий минимум в нуле, равный нулю, и уточнить характер выпуклости по обе стороны от нуля. Исследование закончилось, эскиз графика построен. Для закрепления навыков можно проделать аналогичные вычисления для другой функции, например,

$$y = \sqrt[5]{(x+2)^2(x-3)^3}.$$

В этом случае преобразуем функцию к виду

$$y = x\sqrt[5]{\left(1+\frac{2}{x}\right)^2\left(1-\frac{3}{x}\right)^3}.$$

Сомножители разложим, используя формулу

$$(1+t)^m = 1 + mt + \cdots + \frac{m(m-1)\ldots(m-n+1)t^n}{n!} + o(t^n) \text{ при } t \to 0.$$

Получим

$$x\sqrt[5]{\left(1+\frac{2}{x}\right)^2\left(1-\frac{3}{x}\right)^3} = x\left(1+\frac{4}{5x}+o\left(\frac{1}{x}\right)\right)\left(1-\frac{9}{5x}+o\left(\frac{1}{x}\right)\right).$$

Находим линейную часть разложения, которая и определяет вид наклонной асимптоты $y = x - 1$. Остается вычислить производную, найти критические точки и соответствующие им экстремумы, причем один из них гладкий (в нуле), а другой — не дифференцируемый (в точке минус два). Вторая производная здесь довольно громоздкая, однако вид графика функции уже понятен.

При повторении с выпускниками определения производной функции в точке полезно задание на непосредственное вычисление производной усложнить и выбрать точку, в которой функция не определена. Например, исследуя функцию $y = \frac{\ln x}{1-x}$, сначала доопределим ее при $x = 1$ значением -1, используя замечательный предел. Затем начинаем вычислять производную функции в точке $x = 1$, преобразовывая выражение под знаком предела. Снова используя формулу

$$\ln(1+t) = t - \frac{t^2}{2} + \cdots + (-1)^{n-1}\frac{t^n}{n} + o(t^n) \text{ при } t \to 0, \text{ получим}$$

$$\lim_{x\to 1}\frac{\Delta y}{\Delta x}=\lim_{x\to 1}\frac{1}{x-1}\left(-\frac{\ln\left(1+(x-1)\right)}{x-1}-(-1)\right)=$$
$$=\lim_{x\to 1}\frac{1}{x-1}\left(\frac{(x-1)-\frac{1}{2}(x-1)^2+o\left((x-1)^2\right)}{x-1}-(-1)\right)=\frac{1}{2}.$$

Проводя подобные вычисления для второй производной, получим значение $-\frac{2}{3}$. Материал повторили, осталось закрепить его на новом примере и надеяться, что он будет усвоен учащимися.

Ссылки

1. *Демидович Б. П.* Сборник задач и упражнений по математическому анализу. М. : Наука, 1990.

УДК 372.851

Л. Б. Медведева, Н. Б. Чаплыгина
Ярославский государственный университет им. П. Г. Демидова
E-mail: lbmedvedeva@yandex.ru
E-mail: chaplgn@uniyar.ac.ru

Учебная программа по дополнительной дисциплине «Избранные разделы элементарной математики» для студентов математиков 1-го курса

В статье представлен проект программы коррекционного курса по элементарной математике, призванного ликвидировать пробелы в знаниях студентов-первокурсников.

Библиография: 4 названия.

Ключевые слова: линейные уравнения и системы, неравенства, векторы, тригонометрия, функции.

В 2011/12 и 2013/14 учебных годах для первокурсников математического факультета ЯрГУ были организованы дополнительные занятия по дисциплине «Избранные разделы элементарной математики», востребованность и полезность которой подтверждены практикой (см. [2]). В течение первого семестра студенты повторяли, а некоторые осваивали отдельные разделы элементарной математики, являющиеся опорными на начальном этапе изучения вузовских математических курсов. На этих занятиях предполагалось и оказание консультативной помощи по плановым дисциплинам «Аналитической геометрии», «Алгебре», отдельным вопросам математического анализа. На основе двулетнего опыта преподавания сложилась программа дисциплины, которая рассчитана на 72 аудиторных часа лекций и практических занятий.

Цели освоения дисциплины

Основная задача — ликвидировать пробелы в знаниях элементарной математики, добиться понимания повторяемого материала, сформировать у студентов навыки логических рассуждений, анализа математических ситуаций, доказательств математических утверждений, проверки гипотез, а также навыки самостоятельной работы.

Содержание дисциплины

1. Преобразования выражений, уравнения и системы:

1.1. Решение уравнений и неравенств с модулями, тождественные преобразования.

1.2. Системы линейных уравнений и неравенств. Метод Гаусса решения систем линейных уравнений.

2. Метод математической индукции.

3. Функции:

3.1. Функция. Область определения, область значений, график.

3.2 Свойства функций: четность, ограниченность, периодичность, монотонность, экстремумы, наибольшее и наименьшее значения.

3.3. Основные элементарные функции, их свойства и графики.

3.4. Тригонометрическая окружность. Тригонометрические функции. Обратные тригонометрические функции.

3.5. Понятие сложной и обратной функций.

3.6. Простейшие методы построения графиков функций.

3.7. Многочлены. Корни, разложение на множители, теорема Виета, решение алгебраических уравнений с одной переменной.

4. Векторы:

4.1. Линейные операции над векторами, коллинеарные и компланарные векторы, разложение вектора по заданным векторам.

4.2. Понятие базиса системы векторов и координат вектора в данном базисе. Изменение координат при переходе к новому базису.

4.3. Деление отрезка в данном отношении. Векторный метод решения задач. Задачи на принадлежность трех точек одной прямой и четырех точек одной плоскости.

4.4. Скалярное произведение векторов. Метрические задачи.

4.5. Метод координат на плоскости и в пространстве. Уравнения различных геометрических мест точек.

Как показывает опыт последнего года, лекционные занятия малоэффективны: студенты совершенно не воспринимают ни выводы алгебраических формул, ни доказательства утверждений, ни методы математических рассуждений. Они уверены, что это все лишнее, главное — уметь решать задачи. Однако задачи на доказательство представляют для большинства непреодолимые трудности. Максимум усилий приходится прилагать к тому, чтобы студенты уяснили важные для обучения моменты:

а) математика — это не набор формул и утверждений, используемых для решения задач;

б) знания и умения сами собой не переходят непосредственно от преподавателя к студенту, а усваиваются в процессе выполнения определенной системы действий.

Для этого надо организовать занятия таким образом, чтобы студенты большую часть времени работали самостоятельно или небольшими группами. Это может быть выполнение лабораторной работы, составление опорного констпекта, алгоритма решения задачи, запись доказательства теоремы, написание небольшого математического диктанта и т. п. Понятно, что применение таких форм организации требует соответствующего методического обеспечения.

На наш взгляд, весь материал необходимо разбить на блоки, по каждому из которых должна быть разработана система заданий, отражающих прежде всего теоретическую часть блока. Задания должны содержать теоремы, утверждения и формулы, которые студенты обязаны уметь доказывать и выводить, задачи на доказательство и вычисления. По каждому разделу необходимо предусмотреть контрольное мероприятие: тест, самостоятельную или контрольную работу, индивидуальное собеседование, зачет или мини-экзамен. В представляемой программе разделы и являются таковыми блоками. Ниже представлениы некоторые из них.

Методические материалы по разделу «Векторы»

Вопросы для подготовки к зачету

1. Понятие вектора. Длина, направление, равные векторы, противоположные векторы.

2. Доказать, что $\overrightarrow{AB} = \overrightarrow{DC}$ тогда и только тогда, когда середина отрезка AC совпадает с серединой отрезка BD.

3. Сложение векторов: правила треугольника и параллелограмма и доказательство их равносильности.

4. Свойства сложения векторов (с доказательством).

5. Вычитание векторов. Правила вычитания. Доказать одно из них.

6. Определение операции умножения вектора на число. Построить результат умножения данного вектора $\overrightarrow{a}$ на числа 2, -3, $\frac{3}{2}$, $\sqrt{2}$.

7. Свойства операции умножения вектора на число с доказательством.

8. Понятие коллинеарных векторов. Доказать необходимое и достаточное условие коллинеарности двух ненулевых векторов.

9. Понятие компланарных векторов. Сформулировать и доказать необходимое и достаточное условие компланарности трех векторов.

10. Сформулировать определение линейной комбинации векторов. Привести примеры линейных комбинаций двух, трех векторов.

11. Определения линейно зависимых и независимых векторов. Примеры пар и троек векторов линейно зависимых и независимых.

12. Понятие базиса системы векторов и координат вектора в нем. Доказать теорему о единственности представления каждого вектора пространства линейной комбинацией базисных векторов. Примеры.

13. Определение скалярного произведения векторов, его свойства. Доказать свойство коммутативности и свойство ассоциативности.

14. Вычисление скалярного произведения по координатам векторов-множителей в прямоугольном декартовом базисе с выводом формулы.

15. Понятие аффинной и прямоугольной декартовой системы координат на плоскости и в пространстве. Координаты точки.

16. Решение простейших задач в координатах:

а) нахождение вектора по его начальной и конечной точкам;

б) нахождение середины отрезка и вычисление координат точки деления отрезка в данном отношении по его концам;

в) запись в векторной и координатной формах условий коллинеарности векторов и принадлежности трех точек одной прямой;

г) запись в векторной и координатной формах условий компланарности трех векторов и принадлежности четырех точек одной плоскости.

17. Задачи [2]: №№ 3, 6, 7, 11, 12, 19, 20, 28, 38, 47, 91, 95, 99, 100, 152, 291, 292, 293, 297, 306, 315, 321, 323, 658 – 662, 673, 674, 675, 683.

Опорные конспекты

Опорные конспекты, выдаваемые студентам в начале занятия, сопровождают обзорные лекции. Всего по теме «Векторы» предполагается 2-3 лекции.

Задачи для работы в парах или малых группах

Представлены примеры задач на освоение материала первых шести вопросов программы зачета. Прежде чем студенты начнут работать самостоятельно, решение некоторых типичных задач обсуждается всей группой.

1. В параллелепипеде $ABCDA_1B_1C_1D_1$ точки M и K — середины ребер B_1C_1 и A_1D_1.

1.1 Укажите: а) несколько пар равных векторов;

б) несколько пар противоположных векторов;

в) векторы, отложенные от точек C и B, равные вектору $\overrightarrow{DD_1}$, и векторы, полученные отложением вектора $\overrightarrow{A_1K}$ от точек M, K, B_1.

1.2 Найдите вектор, началом и концом которого являются вершины параллелепипеда, равный алгебраической сумме следующих векторов:

а) $\overrightarrow{AB}+\overrightarrow{A_1D_1}$, б) $\overrightarrow{DC}+\overrightarrow{DA_1}$, в) $\overrightarrow{DA}+\overrightarrow{BB_1}$, г) $\overrightarrow{BC}+\overrightarrow{DB_1}$,

д) $\overrightarrow{AB}-\overrightarrow{DA}+\overrightarrow{AA_1}$, е) $\overrightarrow{A_1B_1}+\overrightarrow{C_1B_1}-\overrightarrow{B_1B}$,

ж) $\overrightarrow{A_1A}-\overrightarrow{D_1A_1}-\overrightarrow{BA}$, з) $\overrightarrow{DD_1}-\overrightarrow{CB}+\overrightarrow{AB}$,

и) $\overrightarrow{AB}+\overrightarrow{CD}+\overrightarrow{DD_1}-\overrightarrow{C_1B_1}$, к) $\overrightarrow{AC}-\overrightarrow{CD}+\overrightarrow{BB_1}-\overrightarrow{C_1A_1}$,

л) $\overrightarrow{AA_1}-\overrightarrow{AB}+\overrightarrow{B_1D1}-\overrightarrow{C_1D_1}-\overrightarrow{CD_1}$,

м) $\overrightarrow{B_1C_1}+\overrightarrow{AB}+\overrightarrow{DD_1}-\overrightarrow{B_1C}+\overrightarrow{BC}-\overrightarrow{AA_1}$.

1.3. Разложите: а) $\overrightarrow{BD_1}$ по векторам $\overrightarrow{BA}$, $\overrightarrow{BC}$, $\overrightarrow{BB_1}$;

б) $\overrightarrow{B_1D_1}$ по векторам $\overrightarrow{AA_1}$, $\overrightarrow{A_1D_1}$, $\overrightarrow{A_1B_1}$;

в) $\overrightarrow{AM}$ по векторам $\overrightarrow{BA}$, $\overrightarrow{BC}$, $\overrightarrow{BB_1}$;

г) $\overrightarrow{KC}$ по векторам $\overrightarrow{AA_1}$, $\overrightarrow{A_1D_1}$, $\overrightarrow{A_1B_1}$.

2. ABC — правильный треугольник, E и F — середины сторон AB и BC. Выразите $\overrightarrow{AB}$, $\overrightarrow{BC}$ и $\overrightarrow{AC}$ через векторы $\overrightarrow{AE} = \overrightarrow{a}$ и $\overrightarrow{AF} = \overrightarrow{b}$.

3. В основании пирамиды $SABCD$ лежит параллелограмм $ABCD$, O — точка пересечения его диагоналей, точки K и L — середины ребер AS и BC соответственно. Разложите векторы $\overrightarrow{AS}$, $\overrightarrow{BL}$, $\overrightarrow{SO}$, $\overrightarrow{KL}$, $\overrightarrow{SK}$, $\overrightarrow{KD}$ по векторам $\overrightarrow{AB} = \overrightarrow{a}$ и $\overrightarrow{BC} = \overrightarrow{b}$, $\overrightarrow{SO} = \overrightarrow{c}$.

Контрольные и самостоятельные работы (примеры)

Самостоятельная работа

1. Даны два треугольника ABC и $A_1B_1C_1$ с центроидами G и G_1. Докажите, что $\overrightarrow{AA_1} + \overrightarrow{BB_1} + \overrightarrow{CC_1} = 3\overrightarrow{GG_1}$.

2. Стороны BC, CA, AB разделены точками A_1, B_1, C_1 (по обходу треугольника) в равных отношениях. Докажите, что треугольники ABC и $A_1B_1C_1$ и имеют общий центроид.

3. На прямой a даны точки M, N, P , а на прямой b – точки A, B, C, причем $MN : NP = AB : BC$. Докажите, что середины отрезков MA, NP, PC лежат на одной прямой.

4. Дан прямой угол ACB. Напишите уравнение множества точек, произведение расстояний от каждой из которых до сторон угла равно разности этих расстояний.

5. Дана окружность с центром в начале координат и радиусом 5 см. Найдите множество середин всех хорд длины 8 см у этой окружности.

Контрольная работа

1. В тетраэдре $ABCD$ точка P делит сторону AB в отношении 2:1, а точка M является центроидом грани ACD . Разложите векторы $\overrightarrow{BM}, \overrightarrow{CP}, \overrightarrow{PM}$ по векторам $\overrightarrow{a} = \overrightarrow{AB}$, $\overrightarrow{b} = \overrightarrow{BC}$, $\overrightarrow{c} = \overrightarrow{AD}$. Запишите координаты указанных векторов в базисе $(\overrightarrow{a}, \overrightarrow{b}, \overrightarrow{c})$.

2. Даны две точки A и B, расстояние между ними равно $2c$. Найдите геометрическое место точек, сумма квадратов расстояний от которых до этих точек равна $2a^2$ при условии, что $a > c$.

3. Даны две точки F_1 и F_2, расстояние между которыми равно $2c$. Найдите геометрическое место точек, абсолютная величина разности расстояний которых до точек F_1 и F_2 равна $2a$ при условии, что $c > a$.

Ссылки

1. *Медведева Л. Б., Чаплыгина Н. Б.* О необходимости занятий по элементарной математике для студентов-математиков 1 курса // Актуальные проблемы совершенствования высшего образования : материалы XII межвуз. науч.-метод. конф. Ярославль : ЯрГУ, 2013.

2. *Моденов П. С. Пархоменко А. С.* Сборник задач по аналитической геометрии. М. : Наука, 1976.

УДК 519.713

М. Л. Мячин
Ярославский государственный университет им. П. Г. Демидова
E-mail: Ltwood@gmail.com

О проблеме интеграции курса КСЕ в учебную программу специальности «Прикладная математика и информатика»

Обсуждаются различные подходы к формированию учебной программы курса «Концепции современного естествознания» для специальности «Прикладная математика и информатика». Описываются базовые принципы, положенные автором в основу курса, читаемого студентам факультета ИВТ.

Библиография: 4 названия.

Ключевые слова: концепции современного естествознания, учебная программа.

Особенности курса и сложившаяся практика. Курс «Концепции современного естествознания» (КСЕ) читается студентам в 4–5 семестрах, т. е. в весеннем семестре 2-го и осеннем семестре 3-го курсов. Первоначально на естественно-научных специальностях этот курс появился в начале 2000-х годов под названием «Математические модели в естествознании» (ММЕ). Затем одновременно с повсеместным введением курса КСЕ (в том числе, и на гуманитарных специальностях) курс ММЕ был просто переименован и превратился в курс КСЕ.

Сама по себе идея курса КСЕ для гуманитарных специальностей состояла в привитии студентам-гуманитариям минимальной естественнонаучной культуры, которой до этого они были совершенно лишены. По причине полного отсутствия у студентов-гуманитариев общей математической грамотности курс КСЕ на гуманитарных специальностях обычно выглядит как обзорный курс философии науки с некоторым уклоном в естественные науки. При этом делается почти безнадежная попытка общими словами передать фактическое содержание основных

естественно-научных теорий. Большинство издаваемых учебников по курсу КСЕ соответствуют именно такому пониманию данной дисциплины.

Подобное выхолащивание курса и превращение его в собрание научно-популярных бесед выглядит бессмысленным и неприемлемым для естественно-научных специальностей и вызвало естественную реакцию в виде появления курсов КСЕ, в большей степени насыщенных фактическим материалом, относящимся к естественным наукам. Студенты, обладающие некоторой математической культурой, оказываются способны усвоить полноценное изложение основных концепций любой естественной науки, но тут появляется проблема, связанная с многообразием естественных наук, представленных в курсе КСЕ. Так, стандартная программа курса предполагает освещение основных концепций физики (включая квантовую механику, общую теорию относительности и космологию), геологии, химии и биологии. При этом автор учебника неизбежно встает перед дилеммой — либо подробно излагать некоторые избранные вопросы в ущерб остальным, либо во всем курсе выдерживать поверхностно-повествовательный стиль изложения.

Учебником КСЕ, ориентированным на студентов-математиков, является книга [1], в которой представлены все перечисленные выше естественные науки, но физика все же доминирует (если изложению основных концепций физики отведено около 100 страниц книги, то оставшиеся 50 страниц вмещают все остальные естественные науки). Отметим все же, что, несмотря на достаточно высокий уровень, из-за широты охвата естественных наук изложение фактического материала в [1] остается научно-популярным и ни один вопрос не изложен достаточно полно.

Поскольку добиться содержательности курса при требуемом широком охвате естественных наук представляется практически невозможным, многие авторы ограничиваются рассмотрением только отдельных вопросов. В книге [2] автор излагает классическую (1/2 объема книги) и статистическую (1/4 объема) механику, уделяя также внимание общим вопросам математического моделирования. Учебное пособие [3] освещает только классическую механику, основы электродинамики и некоторые модели популяционной экологии[2]. В пособии [4] автор ограничивается изложением основных положений математической генетики и нескольких моделей функционирования биологического нейрона.

[2]К сожалению, эта книга содержит множество полемических утверждений, далеких от общепринятых. Так, автор утверждает, что принцип относительности Эйнштейна имеет место только для электромагнитных явлений и неприменим к механическим системам. Такое утверждение выглядит особенно удивительным в то время, когда повсеместно используются системы глобального позиционирования, в которых приходится делать поправку на замедление времени в связанной со спутником движущейся системе координат.

Общая характеристика предлагаемого подхода. Автор данной статьи в течение ряда лет экспериментаровал с различными вариантами построения программы курса КСЕ. Были апробированы, в том числе, и описанные ваше варианты, в которых либо дается одинаково поверхностный обзор многих естественно-научных отраслей, либо относительно подробно излагаются избранные разделы. Для более глубокого изложения мы обычно выбирали механику и теорию колебаний, поскольку студенты 2–3 курсов лучше всего подготовлены к восприятию именно этих областей[3].

В результате оптимальным оказался альтернативный вариант построения курса, в котором программа концентрируется не вокруг отдельных разделов естествознания, а скорее вокруг конкретных естественно-научных задач. При этом сами задачи выбираются таким образом, чтобы в процессе их рассмотрения можно было одновременно дать краткий обзор тех естественно-научных фактов, которые привлекаются для постановки задачи, и проиллюстрировать конкретное применение используемого математического аппарата.

Выбирая задачи, важно руководствоваться основным принципом — ни экскурс в конкретную область естествознания, ни рассказ об используемом математическом аппарате не должен затягиваться и отвлекать внимание студента от сути самой задачи. Это требование автоматически ограничивает нас относительно небольшим кругом задач, относящихся к физике, технике и математической экологии. По аналогичным причинам мы вынуждены использовать только тот математический аппарат, который либо уже знаком студентам, либо непосредственно примыкает к уже изученным ими предметам. Фактически при таком построении курса он почти полностью состоит из тех разделов, которые при наличии достаточного времени непременно вошли бы в курсы алгебры, анализа, дифференциальных уравнений и уравнений математической физики в качестве интересных примеров, демонстрирующих применение изучаемых методов к реальным задачам естествознания[4].

Примерный список тем для курса. В данном разделе мы приведем список тем, которые образуют костяк курса КСЕ, читаемого в настоящее время автором. Все задачи заимствованы из учебной и спе-

[3]В качестве курьеза действующей программы отметим, что курс теории вероятностей изучается на 3-м курсе, так что вплоть до начала 4-го курса (т. е. вообще во всех математических и естественно-научных курсах) нельзя использовать даже простейшие понятия теории вероятностей. Ясно, что этот факт существенного ограничивает выбор изучаемых разделов.

[4]Под давлением нехватки учебных часов из обязательных курсов часто выпадают целые главы. При этом особенно часто страдают геометрические и физические приложения, которые сейчас почти полностью вытеснены из курсов анализа и дифференциальных уравнений.

циальной литературы, но мы не приводим здесь ссылок, поскольку в этом случае список литературы оказался бы неприемлемо длинным.

1. *Задача о размножении в условиях ограниченной кормовой базы.* Здесь, используя элементарные методы исследования фазового портрета, удается рассмотреть задачу об оптимальном режиме использования размножающейся популяции и сделать некоторые нетривиальные выводы относительно неустойчивости экономически оптимальной стратегии.

2. *Задача о вытекании жидкости из сосуда.* На примере этой задачи очень удобно продемонстрировать процесс уточнения математической модели явления. Дело в том, что использование приближенной формулы Торичелли для скорости истекания жидкости при рассмотрении задачи об опустошении сосуда приводит к ряду противоречий и мы приходим к необходимости уточнения исходной формулы.

3. *Задача о динамике уровня водохранилища с плотиной.* В рамках этой простой задачи можно продемонстировать причину возникновения гистерезиса, характерного для многих физических систем.

4. *Математический и физический маятники.* На этом примере можно продемонстрировать перестройку фазового портрета при появлении трения, а для физическаого маятника — зависимость периода нелинейных колебаний от их амплитуды. Для линейного маятника полезно акцентировать внимание на универсальности модели, позволяющей использовать ее для моделирования различных колебательных процессов. Для связанных осцилляторов можно продемонстрировать возникновение биений и понятие нормальных колебаний.

5. *Вынужденные колебания и демпфирование.* Рассмотрение вынужденных колебаний линейного осциллятора позволяет ввести понятие частотной характеристики системы и продемонстрировать имеющие здесь место фазовые эффекты. Рассмотрев колебания в системе с упругой подвеской, можно также продемонстрировать прицип работы резонансного демпфера механических колебаний. Задача является удобным поводом для введения понятия измерительного прибора, для которого удается показать общее соотношение неопределенностей в его спектральной форме. Одновременно можно дать общий обзор обратных задач идентификации систем и понятие о некорректных задачах и методах их решения.

6. *Маятниковые часы Гюйгенса.* Принцип работы маятниковых часов Гюйгенса и их упрощенная математическая модель позволяют продемонстрировать использование точечных отображений для исследования поведения динамической системы. Одновременно полезно показать принцип действия изохронного циклоидального маятника.

7. Описание колебаний двойного маятника. При рассмотрении этой задачи удобно показать переход к инвариантной форме записи механических уравнений движения (уравнения Лагранжа).

8. Параметрическое возбуждение и параметрический резонанс. Описание движения маятника с колеблющейся точкой подвеса позволяет ввести понятия параметрического возбуждения колебаний и продемонстрировать явление параметрического резонанса.

9. Задача о стабилизации перевернутого маятника. Классическая задача П. Л. Капицы о стабилизации верхнего состояния равновесия маятника с колеблющейся точкой подвеса позволяет перейти к рассмотрению систем с управлением и задач, связанных со стабилизацией неустойчивых состояний равновесия.

10. Колебания в ламповом генераторе. Модель лампового генератора позволяет продемонстрировать возникновение автоколебаний и простейшие методы исследования автоколебательных режимов. Одновременно здесь можно показать различие между мягким и жестким режимами возбуждения автоколебаний.

11. Колебания осциллятора с отрицательным трением. Рассмотрение колебаний пространственно-ограниченного осциллятора с отрицательным трением позволяет продемонстрировать возникновение хаотических колебаний в детерминированной механической системе.

12. Задача о сосуществовании популяций. Рассмотрение взаимодействия конкурирующих и симбиотических популяций, а также популяций типа «хищник-жертва» позволяет продемонстрировать простые методы анализа поведения модели путем качественного исследования фазового портрета соответствующей динамической системы. Эти задачи дают также удобный повод для обсуждения понятий грубости и негрубости математической модели.

13. Задача о неустойчивости регулятора Уатта. Эта задача позволяет проиллюстрировать методы, используемые при исследовании систем с управлением, и продемонстрировать проблемы, связанные с неустойчивостью таких систем.

14. Распределенные физические модели. Здесь мы вступаем в область уравнений математической физики, но, по нашему мнению, чрезвычайно полезно как можно раньше показать студентам использование уравнений с частными производными для моделирования явлений в распределенных физических системах, таких как колебания струны и течение вязкой несжимаемой жидкости. Относительно подробное рассмотрение частного решения волнового уравнения, соответствующего бегущей волне, позволяет перейти к изучению явления дисперсии волн.

15. Преломление и отражение электромагнитных волн. Используя простейшие методы исследования уравнений математической физики, мы переходим к рассмотрению уравнений Максвелла для электромагнитного поля. Здесь мы подробно рассматриваем явления преломления и отражения волн на границе сред с различной плотностью (механической или оптической).

16. Инвариантность и симметрия уравнений. Элементарные следствия требования инвариантности уравнений движения можно продемонстрировать уже в рамках классической механики[5]. Рассмотрение вопроса о нарушении инвариантности для уравнений Максвелла по отношению к преобразованию Галилея непосредственно приводит нас к специальной теории относительности. Вывод преобразования Леренца и проверка инвариантности законов Максвелла относительно него позволяет с новой точки зрения взглянуть на классическую электродинамику[6].

Таким образом, теоретический материал нашего курса сконцентрирован вокруг тех задач, которые позволяют проиллюстрировать важнейшие принципы современного естествознания — повсеместное использование динамических систем для моделирования детерминированных эволюционных процессов; идею качественного исследования фазового портрета динамической системы; понятия фазового пространства, фазового потока, фазового портрета и оператора эволюции динамической системы; понятие устойчивости динамической системы и роль устойчивости при использовании математической модели явления; детерминизм и причины недетерминированного поведения динамических систем; понятие грубости динамической системы и различие между грубыми и негрубыми моделями; понятия бифуркации и катастрофы; понятие управления и обратной связи.

Ссылки

1. *Сторожук А. Ю.* Концепции современного естествознания для математиков. Новосибирск, 2008. 167 с.

2. *Юдович В. И.* Математические модели естествознания : курс лекций. СПб. : Лань, 2011. 336 с.

[5]Все механические законы сохранения (а значит, и уравнения движения) могут быть непосредственно получены из условий однородности пространства и времени и изотропности пространства.

[6]Например, мы показываем, что механическое взаимодействие двух проводников с движущимися по ним носителями заряда является существенно релятивистским эффектом, связанным с сокращением пространственных интервалов.

3. *Колесов Ю. С.* Концепции современного естествознания : Математический подход. Ярославль, 2003. 110 с.

4. *Майоров В. В.* Математические модели в естествознании. Ярославль, 2000. 68 с.

УДК 514.17

М. В. Невский
Ярославский государственный университет им. П. Г. Демидова
E-mail: mnevsk@uniyar.ac.ru

О вычислении некоторых характеристик n-мерного симлекса

Рассматриваются задача о вычислении максимальноого в симплексе отрезка данного направления, а также некоторые приложения.

Библиография: 3 названия.

Ключевые слова: n-мерный симплекс, отрезок, максимальная длина, гомотетия.

Описание используемых результатов

Пусть S — невырожденный симплекс в $\mathbb{R}^n$ и v — ненулевой n-мерный вектор. Обозначим через $d^v(S)$ максимальную длину отрезка, принадлежащего S и параллельного v. Рассмотрим задачу о вычислении величины $d^v(S)$ и концов максимального отрезка по координатам v и вершин S. В докладе приводятся соответствующие формулы и примеры их использования.

Обозначим через $x^{(j)} = \left(x_1^{(j)}, \ldots, x_n^{(j)}\right)$ $(1 \le j \le n+1)$ вершины S. Пусть $\lambda_1(x), \ldots, \lambda_{n+1}(x)$ — барицентрические координаты точки $x \in \mathbb{R}^n$ относительно S. Они определяются свойствами

$$\sum_{j=1}^{n+1} \lambda_j(x) x^{(j)} = x, \quad \sum_{j=1}^{n+1} \lambda_j(x) = 1. \tag{1}$$

Симплекс S задаётся неравенствами $\lambda_j(x) \ge 0$; точки его $(n-1)$-мерных граней удовлетворяют уравнениям $\lambda_j(x) = 0$. Рассмотрим матрицу

$$\mathbf{A} := \begin{pmatrix} x_1^{(1)} & \ldots & x_n^{(1)} & 1 \\ x_1^{(2)} & \ldots & x_n^{(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(n+1)} & \ldots & x_n^{(n+1)} & 1 \end{pmatrix}.$$

Положим $\mathbf{A}^{-1} = (l_{ij})$. Тогда $\lambda_j(x) = l_{1j}x_1 + \ldots + l_{nj}x_n + l_{n+1,j}$.

Обозначим через $v_1, \ldots, v_n$ координаты данного вектора v. Введём в рассмотрение числа $(1 \leq j \leq n+1)$

$$m_j := \sum_{k=1}^{n} l_{kj} v_k, \tag{2}$$

$$\alpha_j := \frac{|m_j| - m_j}{\sum\limits_{k=1}^{n+1} |m_k|}, \quad \beta_j := \frac{|m_j| + m_j}{\sum\limits_{k=1}^{n+1} |m_k|}. \tag{3}$$

Через $\|\cdot\|$ обозначим евклидову норму в $\mathbb{R}^n$. В статье [1] автор доказал следующее утверждение.

Теорема. *Справедливо равенство*

$$d^v(S) = \frac{2\|v\|}{\sum\limits_{j=1}^{n+1} |m_j|}. \tag{4}$$

Концы единственного отрезка максимальной длины, принадлежащего S и параллельного v, суть точки

$$a = \sum_{j=1}^{n+1} \alpha_j x^{(j)}, \quad b = \sum_{j=1}^{n+1} \beta_j x^{(j)}. \tag{5}$$

Интересно, что максимальный в S отрезок, параллельный вектору v, однозначно характеризуется следующим условием: каждая $(n-1)$-мерная грань S содержит хотя бы один из концов этого отрезка. Эта характеризация также установлена в [1].

Сформулируем утверждения, которые выводятся из приведённой выше теоремы с привлечением некоторых других результатов автора. Пусть $\Sigma(S; v)$ есть $(n-1)$-мерная мера проекции симплекса S на гиперплоскость, ортогональную вектору v.

Следствие 1. *Имеют место равенства*

$$\Sigma(S; v) = \frac{n \cdot \mathrm{vol}(S)}{d^v(S)} = \frac{|\det(\mathbf{A})|}{2(n-1)!\|v\|} \sum_{j=1}^{n+1} |m_j|\,. \tag{6}$$

Через σS обозначим образ S при гомотетии с коэффициентом σ и центром гомотетии в центре тяжести S. Пусть V — невырожденный параллелепипед в $\mathbb{R}^n$, ребра которого задаются линейно независимыми векторами $v^{(1)}, \ldots, v^{(n)}$. Через $\alpha(V; S)$ обозначим минимальное

$\sigma > 0$ такое, что V содержится в трансляте симплекса σS. Вычислим величину

$$M := \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n+1} \left| \sum_{k=1}^{n} l_{kj} v_k^{(i)} \right|. \tag{7}$$

Следствие 2. *Справедливы равенства*

$$\alpha(V; S) = \sum_{i=1}^{n} \frac{\left\| v^{(i)} \right\|}{d^{v^{(i)}}(S)} = M.$$

Следствие 3. *Неравенство $M \leq 1$ эквивалентно тому, что V содержится в трансляте симплекса S. Равенство $M = 1$ эквивалентно тому, что некоторый транслят S' симплекса S описан вокруг V (т. е. $V \subset S'$ и каждая $(n-1)$-мерная грань S' содержит вершину V).*

Изложенные задачи могут быть с пользой для студентов освещены в учебном процессе. Отметим, что к данной тематике относятся и доклады автора на двух предыдущих научно-методических конференциях "Преподавание математики и компьютерных наук в классическом университете" (см. [2], [3]).

Пример

Ограничимся здесь плоским случаем ($n = 2$), когда S представляет собой треугольник. Именно, пусть S есть треугольник с вершинами $x^{(1)} = (1, 0)$, $x^{(2)} = \left(\frac{1}{2}, 1\right)$, $x^{(3)} = (0, 0)$. В данном случае

$$\mathbf{A} := \begin{pmatrix} 1 & 0 & 1 \\ \frac{1}{2} & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{A}^{-1} = \begin{pmatrix} 1 & 0 & -1 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix}.$$

Значит, $\lambda_1(x) = x_1 - \frac{1}{2}x_2$, $\lambda_2(x) = x_2$, $\lambda_3(x) = -x_1 - \frac{1}{2}x_2 + 1$. Напомним, что числа l_{ij} составляют матрицу $\mathbf{A}^{-1}$ и являются коэффициентами многочленов λ_j. Студентам полезно убедиться, что равенства (1) действительно имеют место.

Пусть дан двумерный вектор $v = (1, 1)$. Вычислим величину максимального в S отрезка, параллельного v, и координаты его концов. В нашем случае $v_1 = v_2 = 1$. Вычисления по формулам (2), (3) дают

$$m_1 = \frac{1}{2}, \quad m_2 = 1, \quad m_3 = -\frac{3}{2},$$

$$\alpha_1 = \alpha_2 = 0, \quad \alpha_3 = 1, \quad \beta_1 = \frac{1}{3}, \quad \beta_2 = \frac{2}{3}, \quad \beta_3 = 0.$$

Так как $\|v\| = \sqrt{2}$, то по формуле (4) имеем

$$d^v(S) = \frac{2\|v\|}{\sum\limits_{j=1}^{n+1} |m_j|} = \frac{2\sqrt{2}}{3}.$$

В соответствии с (5) единственный максимальный в S отрезок, параллельный v, имеет вид $I = [a, b]$, где

$$a = x^{(3)} = (0, 0), \quad b = \frac{1}{3}x^{(1)} + \frac{2}{3}x^{(3)} = \left(\frac{2}{3}, \frac{2}{3}\right).$$

Теперь найдём $(n-1)$-меру (иначе говоря, длину) $\Sigma(S; v)$ ортогональной проекции S на прямую, ортогональную v (т. е. прямую $x_1 + x_2 = 1$). Так как $\det(\mathbf{A}) = 1$, то по формуле (6) длина проекции S на указанную прямую равна

$$\Sigma(S; v) = \frac{|\det(\mathbf{A})|}{2(n-1)!\|v\|} \sum_{j=1}^{n+1} |m_j| = \frac{1}{2\sqrt{2}} \cdot 3 = \frac{3\sqrt{2}}{4}.$$

В справедливости полученных результатов можно убедиться непосредственно по чертежу, проделав несложные вычисления.

Наконец, для того же треугольника S рассмотрим параллелограмм V, одна вершина которого есть $(0, 0)$, а рёбра, исходящие из этой вершины, задаются векторами $v^{(1)} = (1, 0), v^{(2)} = \left(\frac{1}{2}, 1\right)$. Применение (7) и следствия 2 даёт $\alpha(V; S) = 2$. Этот результат также может быть проверен непосредственно. Действительно, обозначим через S' треугольник с вершинами $(2, 0), (2, 1), (0, 0)$. Нетрудно видеть, что S' — удвоенный гомотетический образ S, причём S' описан вокруг V (т. е. $V \subset S'$ и каждая сторона S' содержит вершину V). Из этих условий и следует, что $\alpha(V; S) = 2$.

Ссылки

1 *Невский М. В.* Вычисление максимального в симплексе отрезка данного направления // Фундаметальная и прикладная математика. 2013. Т. 18, вып. 2. С. 147–152.

2. *Невский М. В.* О некоторых свойствах базисных многочленов Лагранжа // Преподавание математики и компьютерных наук в классическом университете : материалы 3-й научно-методической конференции преподавателей математического факультета и факультета информатики и вычислительной техники Ярославского государственного университета им. П. Г. Демидова. Ярославль, 2010. С. 111–117.

3. *Невский М. В.* О некоторых задачах, связанных с осевыми диаметрами // Преподавание математики и компьютерных наук в классическом университете : материалы 4-й научно-методической конференции преподавателей математического факультета и факультета информатики и вычислительной техники Ярославского государственного университета им. П. Г. Демидова. Ярославль, 2010. С. 66–68.

УДК 512.512(075)

Ф. И. Папоркова
Ярославский государственный университет им. П. Г. Демидова
E-mail: florida@uniyar.ac.ru

Проблемы проведения практических занятий

Приводятся примеры заданий, способствующих формированию у студентов элементов исследовательской компетенции.
Библиография: 3 названия.

Ключевые слова: прямая, плоскость, задачи с параметрами.

Смена ценностных ориентаций в обществе, введение в действие ЕГЭ и ФГОС — новых стандартов образования с каждым годом вызывают все большие сложности и проблемы в учебном процессе. Остановимся на некоторых из них. Так, практические занятия по курсу «Алгебра и геометрия» всегда сопровождал основной сборник задач по аналитической геометрии П. С. Моденова и А.С. Пархоменко [1]. Традиционный стандартный набор задач по основным темам данного курса из этого сборника всегда был по силам основной массе студентов. Легко решались даже задачи повышенной сложности. В качестве примера можно привести

Задача 198 * [1]. Даны три некомпланарных вектора $a = \{x_1, y_1, z_1\}$, $b = \{x_2, y_2, z_2\}$, $n = \{A, B, C\}$. Найти площадь параллелограмма, являющегося ортогональной проекцией на плоскость, перпендикулярную к вектору n, параллелограмма, построенного на векторах a и b.

Для решения этой задачи достаточно было знать школьную формулу, что объем наклонного параллелепипеда равен произведению длины бокового ребра на площадь сечения, перпендикулярного данному ребру. В настоящий период такого рода задачи можно предлагать лишь избранным студентам и только в качестве домашнего задания с подробными комментариями. С большим интересом решались задачи с параметрами из сборника под редакцией Д. В. Беклемишева [2], по теме "Прямая в пространстве". Например:

Задача 6.24 [2]. При каких a прямая

$$\frac{x}{1} = \frac{y}{a} = \frac{z-2}{-1} :$$

1) пересекает плоскость $3a^2x + ay + z - 4a = 0$;
2) параллельна этой плоскости;
3) лежит в этой плоскости?

Задача 6.26 [2]. При каких a прямые

$$\frac{x-1}{a} = \frac{y-1}{1} = \frac{z-(a-2)^2}{a}, \qquad \frac{x}{1} = \frac{y}{a} = \frac{z}{1}$$

1) пересекаются; 2) скрещиваются; 3) параллельны; 4) совпадают?

На данном этапе задачи с параметрами (даже простейшего типа) понимают и решают лишь единицы. В такой ситуации возникла необходимость заказать в библиотеке дополнительный задачник Д. В. Клетеника, в котором имеются теоретические введения ко всем разделам курса. Но даже такие изменения не привели к заметному повышению успеваемости основной группы студентов.

Аналогичная картина наблюдается во втором семестре при изучении второй части объединенного курса аналитической геометрии и линейной алгебры.

Например, по теме «Линейный оператор» студентам предлагается стандартный набор заданий из классического сборника задач под редакцией И. В. Проскурякова [3, №№ 1441 – 1458], решение которых способствует усвоению основных понятий. Однако, если раньше наряду с задачами типа:

1. Оператор A переводит вектор $x = (x_1, x_2, x_3)$ в вектор

   $$A(x) = (x_2 - x_1, x_2, x_3 - 2x_2 + 3x_1).$$

   Показать, что A — линейный оператор и найти его матрицу;

2. Пусть линейный оператор A в пространстве R^3 имеет в базисе $((8, -6, 7), (-16, 7, -13), (9, -3, 7))$ матрицу

   $$M_A = \begin{pmatrix} 1 & -18 & 15 \\ -1 & -22 & 20 \\ 1 & -25 & 22 \end{pmatrix} \qquad (1)$$

   Найти его матрицу в базисе $((1, -2, 1), (3, -1, 2), (2, 1, 2))$;

3. Оператор A переводит вектор $x = (x_1, x_2, x_3)$ в вектор

   $$A(x) = (x_1, x_2 - 3x_3, x_3 + 2x_1).$$

   Найти ядро, образ и собственные векторы оператора

студенты также достаточно легко осваивали решение задач, в которых линейный оператор действует в пространстве многочленов степени, не превосходящей $n, (n \leq 4)$, то теперь это вызывает трудности у большей части студентов.

Несмотря на все возникающие трудности, связанные с восприятием и пониманием студентами материала, приведем пример цикла задач, позволяющий сохранить высокий научно-методический уровень проведения практических занятий и достичь желаемых результатов по развитию творческого мышления и элементов исследовательской компетенции у студентов, необходимых для решения задач определенной сложности.

Задача. В пространстве R^3 с базисом $((1,0,0),(0,1,0),(0,0,1))$ найти матрицу оператора ортогонального проектирования P на подпространство L, если L есть:

1. i)плоскость $z = 0$; ii) плоскость $x = 0$;
2. прямая $x = z = 0$;
3. прямая $x = y = z$;
4. плоскость $x + 2y + 3z = 0$;
5. плоскость, натянутая на векторы $a = \{1, -2, 1\}, \quad b = \{3, 2, -3\}$;

Определить ядро и образ.
В случае плоскости $x + 2y + 3z = 0$ указать базис, в котором матрица оператора будет иметь вид:

$$M_A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Задача. Определить, могут ли матрицы

$$\begin{pmatrix} 0 & 3 & -1 \\ 2 & 1 & 0 \\ -1 & 0 & 4 \end{pmatrix}, \qquad \begin{pmatrix} 3 & 3 & 3 \\ 1 & 1 & 1 \\ 0 & 2 & 4 \end{pmatrix} \tag{3}$$

быть матрицами одного линейного оператора в разных базисах.

Ссылки

1. *Моденов П. С., Пархоменко А.С.* Сборник задач по аналитической геометрии. М. : Dinamics, 2002.

2. *Беклемишева Л. А., Петрович А. Ю., Чубаров И. А.* Сборник задач по аналитической геометрии и линейной алгебре. М. : Физматлит, 2003.

3. *Проскуряков И. В.* Сборник задач по линейной алгебре, М. : Юнимедиастайл, 2002.

УДК 378.147

В. С. Рублев
Ярославский государственный университет им. П. Г. Демидова
E-mail: roublev@mail.ru

Организация учебной практики по информатике и программированию для студентов специальности Фундаментальная информатика и информационные технологии

Обсуждаются вопросы организации учебной практики по информатике и программированию.

Библиография: 2 названия.

Ключевые слова: информатика, программирование, учебная практика, организация.

Цели и задачи практики

Учебная практика предназначена для закрепления и углубления знаний и умений, полученных студентами в течение года по программированию и таким стуктурам дискретной математики, как множества и булевы функции.

Одним из недостатков обучения программированию студентов 1-го курса является перенос основного внимания на кодирование программы в ущерб тщательной разработке алгоритма. Многие студенты сразу пишут программу на компьютере, не разработав и не проверив ее алгоритм, надеясь исправить ошибки в процессе отладки. В результате процесс поиска логических ошибок нередко неоправданно затягивается, да и алгоритм программы часто содержит неэффективную реализацию.

Заметим также, что в процессе разработки программного обеспечения в программистских организациях участвуют специалисты трех профессий:

- **алгоритмист**, который разрабатывает структуру алгоритма, пошаговый алгоритм каждой процедуры и всего алгоритма, полный набор тестов для проверки алгоритма и проверяет его тестированием этих тестов; он должен иметь высокое образование, от которого зависит успех всего проекта;

- **кодировщик**, который по алгоритму пишет программный код на выбранном языке программирования; он его также тестирует подготовленными алгоритмистом тестами; чаще всего от него требуется лишь среднее образование;

- **тестировщик**, в задачу которого входит тщательная проверка программы; с этой целью он пытается обнаружить тесты, на которые программа не рассчитана; от него требуется высокое логическое мышление, а следовательно, высшее образование.

Поэтому одной из целей практики должно являться освоение всего цикла разработки программного продукта с упором на качественную разработку алгоритма и его тестирования до кодирования и отладки программы.

Знания, полученные студентами по темам *Множества* и *Булевы функции*, могут быть использованы при разработке алгоритмов и программ. Однако реализация множества в виде массива не является эффективной ни в плане используемой для этого памяти, ни в плане вычислительной трудоемкости операций со множествами. Использование булевых функций позволяет в сложных случаях проектировать эффективные конструкции условий циклов и условных операторов. Поэтому другой целью практики должно являться освоение эффективной организации множеств и операций над ними, булевых функций и их анализ с помощью программного получения таблиц истинности и программного их анализа.

Задачами практики являются:

1. Освоение всех этапов разработки программного обеспечения:

   a) разработка полного набора тестов для задачи;

   b) разработка неформального пошагового описания алгоритма;

   c) тестирование пошагового описания алгоритма;

   d) кодирование алгоритма программы и ее частей;

   e) тестирование программы.

2. Освоение программной реализации множеств.

3. Освоение программной реализации булевых функций.

В связи с этим каждый студент получает 3 индивидуальных задания:

1. Разработка процедуры сложного преобразования матрицы.

2. Класс множеств и программное тестирование утверждения для множеств.

3. Булевы функции и программирование таблиц истинности.

Примеры индивидуальных заданий

В процессе выполнения первого задания предполагается, что каждый студент пройдет начальное освоение всех этапов, связанных с созданием алгоритма и программного продукта. Разработка первых трех этапов ведется на бумаге последовательно по этапам: к следующему этапу следует переходить только после полной проверки предыдущего этапа преподавателем и устранения ошибок, если таковые были отмечены.

В качестве учебной задачи разработки программного обеспечения взято задание разработки процедуры сложного преобразования матрицы. Примером может быть следующее задание:

Квадратная целочисленная матрица A размерности k^2 ($k>0$) состоит из k^2 непересекающихся квадратных подматриц B_{ij} ($i=1,\ldots,k$; $j=1,\ldots,k$) размерности k. В каждой подматрице B_{ij}, не содержащей минимальный элемент матрицы A, переворачивается вокруг горизонтальной оси симметрии подматрицы ее внутренность (элементы, не содержащиеся в крайних строках или столбцах), а в самой матрице A поворачивается на 180^0 часть контура из подматриц (подматрицы, содержащие крайние строки или столбцы матрицы A), состоящая только из преобразованных подматриц.

Для выполнения второго задания студент изучает класс множеств и реализацию методов класса. При этом не все требующиеся для программы методы могут быть описаны. Студенту необходимо дополнить класс необходимыми методами и их реализацией. В качестве индивидуального задания берется задача 1 из индивидуального задания 1 по дискретной математике (ДМ) [1], в которой дано некоторое утверждение для трех любых подмножеств произвольного универсального множества. Студент должен разработать программу проверки этого утверждения на тестовых числовых примерах, подготовленных в порядке выполнения задания 1 по ДМ. Примером является задание: Для следующего утверждения для множеств

$$X_1 \cap X_2 \cap X_3 = \emptyset \text{ и } X_1 \cup X_2 \cup X_3 = U \leftrightarrow (X_2 \setminus X_3) \cup (X_1 \setminus X_2) = X_1 \cup \overline{X}_3$$

разработать программу проверки этого утверждения с помощью тестов для множеств из целочисленных неотрицательных элементов.

Для выполнения третьего задания в качестве индивидуального задания берется задача 1 из индивидуального задания 4 по ДМ [2], где требуется провести доказательство утверждения для множеств путем сведения этой задачи к задаче проверки тождественной истинности булевой функции. Образцы соответствующих программ даны. Но, кроме этого, требуется написать программу вывода таблицы истинности для булевой функции, соответствующей отношению множеств (задача 1 индивидуального задания 5 по ДМ [2]).

График выполнения заданий практики

График выполнения заданий предполагает понедельные сроки выполнения заданий и их этапов (срок начинается в учебный день практики на неделе и заканчивается через 6 дней включительно). Для каждого этапа студент сдает отчет по этой части этапа и после приема его преподавателем может перейти к выполнению следующего этапа.

Студенты, сдавшие все этапы и задания не позже срока, получают оценку **отлично** по зачету практики. Студенты, сдавшие все этапы и задания до конца семестра, получают оценку **хорошо** по зачету практики. Студенты, сдавшие все этапы и задания на перезачете (после конца семестра и в начале следующего семестра), получают оценку **удовлетворительно** по зачету практики. Студенты, не сдавшие перезачет, не аттестуются.

График заданий и их этапов

2-я неделя – разработка полного набора тестов для задания 1.

4-я неделя – разработка неформального пошагового описания алгоритма.

6-я неделя – тестирование пошагового описания алгоритма.

8-я неделя – кодирование алгоритма программы и его частей.

9-я неделя – тестирование программы; сдача полного отчета по заданию 1.

11-я неделя – сдача отчета по заданию 2.

12-я неделя – сдача отчета по первому и второму этапам задания 3.

13-я неделя – выполнение этапа 3 и сдача полного отчета по заданию 3.

Ссылки

1. *Рублев В.С.* Множества (индивидуальная работа № 1 по дисциплине «Основы дискретной математики») : методические указания. Ярославль: ЯрГУ, 2009. 36 с.

2. *Рублев В.С.* Булевы функции (индивидуальные работы № 4, 5 по дисциплине «Основы дискретной математики») : методические указания. Ярославль: ЯрГУ, 2009. 48 с.

УДК 004.382.3; 004.727

А. В. Соколов
Ярославский государственный университет им. П. Г. Демидова
E-mail: abc@uniyar.ac.ru

Технология виртуализации в учебном практикуме компьютерных дисциплин

Обсуждаются вопросы, связанные с использованием технологии виртуализации при преподавании компьютерных дисциплин.

Библиография : 1 название.

Ключевые слова: виртуализация, облачные вычисления.

Термин «виртуализация» в компьютерных технологиях используется очень широко[1]. Однако в зависимости от контекста его значение (и само назначение виртуализации) сильно меняется. Можно выделить два основных направления – виртуализация платформ и ресурсов. В первом случае мы получаем виртуальные машины – некие программные абстракции, запускаемые на платформе реальных аппаратно-программных систем («физических» серверов). Во втором случае выполняется объединение (или, наоборот, разбиение) физических ресурсов на логические (под ресурсами понимаются процессоры, дисковое пространство, сетевые адаптеры). Основные задачи, которые решаются с помощью технологий виртуализации, – это:

- объединение серверов для повышения коэффициента полезного использования аппаратуры (процессора, дискового пространства, сети);
- разработка и тестирование программного обеспечения (ПО); разработчик имеет возможность на одном компьютере иметь несколько машин с разными операционными системами (ОС) или имеющими разную конфигурацию;
- поддержка старых или тестирование новых ОС в организации;
- изолирование потенциально-опасного окружения;

[1]URL : http://ru.wikipedia.org/wiki/Виртуализация

- перенос служб, сервисов на новую платформу при отказе оборудования или плановых технических мероприятиях;
- и многое другое (см., например, [1]).

Часть задач касается служб поддержки компьютерной сети организации, но многое имеет прямое отношение к построению учебного процесса для компьютерных дисциплин. Отметим, что из перечисленного многое уже доступно даже для систем виртуализации для персональных компьютеров (это, к примеру, программные пакеты VMWare Workstation, Microsoft Virtual PC, Oracle Virtual Box) и, таким образом, может и используется учащимися для самостоятельной (домашней) работы. Информация по этим программным продуктам доступна на специализированных сайтах[2]. В чем заключается необходимость использования технологии виртуализации в наших учебных лабораториях? Не существует возможности установить бесконфликтно на компьютерах этих лабораторий все ПО, используемое в многочисленных учебных курсах (это же относится и к установке нескольких ОС на один компьютер). Более того, часть ПО потенциально опасна для локальной сети, часть задач (например, администрирование ОС) приводит к потере работоспособности машин и т. д. Таким образом, в ближайшее время планируется создание рабочих мест для изучения ОС и серверных служб, их администрирования, для разработки сетевых приложений, для моделирования распределенных вычислительных систем. Во всех этих случаях виртуальные машины предоставляют огромные возможности для учебного процесса. Для внедрения указанной технологии в учебный процесс в учебном корпусе 7 разворачивается инфраструктура виртуальных столов (VDI, virtual desktop infrastructure). Инфраструктура виртуализации включает в себя несколько так называемых «мониторов виртуальных машин» (Virtual Machine Monitor), иначе – гипервизоров. Это физические серверы, на которых и создаются, клонируются, перемещаются по сети виртуальные машины. Управляющая система VDI (к использованию выбрана система от фирмы Oracle) предоставляет пользователям «вход в мир виртуальных машин», занимается задачами по созданию сеансовых виртуальных машин из имеющихся шаблонов (образов) и, конечно, дает администраторам возможность готовить, исправлять/обновлять образы, распределять ресурсы, регулировать права доступа пользователей. В инфраструктуру входят также сетевые хранилища данных – дисковые массивы емкостью в десятки терабайт и высокоскоростное сетевое коммутирующее оборудование. Соединение с системой VDI (т. е. получение виртуального рабочего стола) осуществляется по сети с любого компьютера или мобильно-

[2]URL : http://www.vmware.com/ru/, URL : http://www.virtualbox.org/, URL : http://www.microsoft.com/virtualization/ru/ru/

го устройства. Программа-клиент имеется практически для любых ОС. На данной технологической основе будет создан репозиторий готовых к использованию виртуальных машин с различными гостевыми операционными системами, с отличающимися от базового набора комплектами ПО. Важно, что с такими машинами пользователь может работать в режиме администратора и подвергать их любым экспериментам – ведь виртуальные машины создаются для него на время одного сеанса, на время одного занятия. Таким образом, перед факультетами встает задача внедрения указанной технологии в учебный процесс, наступает этап разработки учебных практикумов, заключающийся в создании образов виртуальных машин, сконфигурированных под конкретную учебную цель, под определенное задание. Поскольку образ виртуальной машины представляет собой файл (хоть и большой), он может быть использован учащимися в качестве методического пособия для самостоятельной подготовки, путем копирования на внешний носитель (например, USB). Создание виртуальной машины не составит особой трудности как преподавателям, так и студентам. Вышеуказанное ПО для персональных компьютеров является или открытым, как Oracle Virtual Box, или доступным нашим сотрудникам/студентам в рамках факультетской академической подписки – как VMWare Workstation.

Ссылки

1. *Matthew Portnoy.* Virtualization Essentials. Indianopolis, Indiana : John Wiley & Sons, 2012.

УДК 517.52

Е. А. Тимофеев
Ярославский государственный университет им. П. Г. Демидова
E-mail: timofeevea@gmail.com

Суммирование гармонических рядов

В статье описывается метод асимптотического анализа одного класса сумм, возникающих в комбинаторной математике, дискретных вероятностных моделях и алгоритмах. Этод метод основан на применении преобразования Меллина.

Библиография: 2 названия.

Ключевые слова: гармонические суммы, преобразование Меллина, асимптотика.

Сумма вида

$$f(x) = \sum_k \lambda_k \phi(\nu_k x) \tag{1}$$

называется *гармонической*. Будем предполагать, что величины $\nu_k \to 0$ при $k \to \infty$.

Для нахождения $f(x)$ будем применять преобразование Меллина

$$F(z) = \int_0^\infty f(x) x^{z-1} dx. \tag{2}$$

Уравнение (1) принимает вид

$$F(z) = \Lambda(z)\, \Phi(z), \tag{3}$$

где $\Phi(z)$ – преобразование Меллина [1] функции $\phi(x)$ и

$$\Lambda(z) = \sum_k \lambda_k \nu_k^{-z}. \tag{4}$$

Для существования преобразования Меллина на функцию $\phi(x)$ нужно наложить ограничения. Далее будем считать, что

1) $\phi(x)$ ограничена на каждом отрезке луча $(0, \infty)$;

2) $\phi(x) = O(x^{-a})$ при $x \to 0$.

3) $\phi(x) = O(x^{-b})$, $a < b$, при $x \to \infty$;

В этом случае в полосе $a < \Re z < b$ функция $\Phi(z)$ будет аналитической.

В случае конечных сумм этого достаточно. Для бесконечных сумм нужно дополнительное требование:

4) ряд (4) сходится в полуплоскости $\Re z > \sigma$, где $\sigma < b$.

Стандартный подход к определению асимптотики $f(x)$ состоит в нахождении обратного преобразования Меллина

$$f(x) = \frac{1}{2\pi i} \int\limits_{\sigma-i\infty}^{\sigma+i\infty} F(z)x^{-z}\,dz, \tag{5}$$

для некоторого σ.

Для мероморфных функций $F(z)$ задача нахождения обратного преобразования Меллина сводится к вычислению вычетов функции $F(z)x^{-z}$.

При $x \to 0$

$$f(x) \sim \sum_{z_0 \in P} \mathsf{Res}\left(F(z)x^{-z}, z_0\right),$$

где P – множество полюсов в полуплоскости $\Re z < \sigma$, $a < \sigma$.

При $x \to \infty$

$$f(x) \sim -\sum_{z_0 \in P} \mathsf{Res}\left(F(z)x^{-z}, z_0\right),$$

где P – множество полюсов в полуплоскости $\Re z > \sigma$, $\sigma < b$.

Пример 1. Найти асимптотику функции при $x \to 0$.

$$f_1(x) = \sum_{k=1}^{\infty} e^{-kx}. \tag{6}$$

В полосе $0 < \Re z < \infty$ преобразованием Меллина функции $\phi(x) = e^{-x}$ является гамма-функция, а

$$\Lambda(z) = \zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z},$$

поэтому

$$F_1(z) = \zeta(z)\Gamma(z).$$

У функции $\zeta(z)$ единственный полюс в точке $z = 1$ с вычетом равным 1, $\zeta(0) = -1/2$, а функция $\Gamma(z)$ имеет простые полюса в точках $z = -k$ $(k = 0, 1, \dots)$, которым соответствуют вычеты $\frac{(-1)^k}{k!}$. Выберем $\sigma = 2$. Следовательно, при $x \to 0$

$$f_1(x) \sim \frac{1}{x} - \frac{1}{2} + \sum_{k=1}^{\infty} \frac{(-1)^k}{k!}\zeta(-k)x^k.$$

Это разложение является общеизвестным определением чисел Бернулли

$$\frac{x}{e^x-1}=\sum_{k=0}^{\infty}B_k\frac{x^k}{k!}.$$

Сравнивая это определение с полученным разложением $f_1(x)$, получим

$$\zeta(-k)=-\frac{B_{k+1}}{k+1} \tag{7}$$

для $k\geq 1$, в частности, $\zeta(-k)=0$ для всех четных $k\geq 1$, поскольку $B_m=0$ для всех нечетных $m>1$.

Пример 2. Найти асимптотику функции при $x\to 0$.

$$f_1(x)=\sum_{k=1}^{\infty}e^{-\sqrt{k}\,x}. \tag{8}$$

В полосе $0<\Re z<\infty$ преобразованием Меллина функции $\phi(x)=e^{-x}$ является гамма-функция, а $\Lambda(z)=\zeta(z/2)$, поэтому

$$F_2(z)=\zeta(z/2)\Gamma(z).$$

У функции $\zeta(z)$ единственный полюс в точке $z=1$ с вычетом, равным 1, $\zeta(0)=-1/2$, а функция $\Gamma(z)$ имеет простые полюса в точках $z=-k$ $(k=0,1,\dots)$, которым соответствуют вычеты $\frac{(-1)^k}{k!}$. Выберем $\sigma=4$. Следовательно, при $x\to 0$

$$f_2(x)\sim\frac{2}{x^2}-\frac{1}{2}+\sum_{k=1}^{\infty}\frac{(-1)^k}{k!}\zeta\left(-\frac{k}{2}\right)x^k.$$

Пример 3. Формула Стирлинга. Применим разложение гамма-функции в произведение [2, формула 8.322]

$$f_3(x)=\ln\Gamma(x+1)+\mathcal{C}x=\sum_{k=1}^{\infty}\left[\frac{x}{k}-\ln\left(1+\frac{x}{k}\right)\right], \tag{9}$$

где $\mathcal{C}$ – константа Эйлера.

В полосе $-2<\Re z<-1$ для функции $\phi(x)=x-\ln(1+x)$ преобразование Меллина существует [2, формула 4.293.3] и

$$\Phi(z)=-\frac{\pi}{z\sin\pi z}.$$

Поскольку $\Lambda(z)=\zeta(-z)$,

$$F_3(z)=-\frac{\pi}{z\sin\pi z}\zeta(-z).$$

Функция $F_3(z)$ имеет двойные полюса в точках $z=-1$, $z=0$ и простые полюса в остальных целых точках. Выберем $\sigma=-3/2$. Найдя вычеты в точках полуплоскости $\Re z>\sigma$, получим

$$\ln\Gamma(x+1)\sim x\ln x-x+\frac{1}{2}\ln x-\zeta'(0)+\sum_{n=1}^{\infty}\frac{(-1)^{n-1}\zeta(-n)}{n(n-1)x^n}.$$

Подставляя $\zeta'(0)=-\ln\sqrt{2\pi}$, $\zeta(-2m)=0$, $\zeta(-2m+1)=-\dfrac{B_{2m}}{2m}$, получим

$$\ln\Gamma(x+1)\sim x\ln x-x+\frac{1}{2}\ln x+\frac{1}{2}\ln(2\pi)+\sum_{n=1}^{\infty}\frac{B_{2n})}{2n(2n-1)x^{2n-1}}.$$

Пример 4. Найти асимптотику при $x\to\infty$

$$f_4(x)=\sum_{k=0}^{\infty}\left(1-e^{-x2^{-k}}\right). \tag{10}$$

В полосе $-1<\Re z<0$ для функции $\phi(x)=1-e^{-x}$ преобразование Меллина существует и

$$\Phi(z)=-\Gamma(z).$$

Поскольку

$$\Lambda(z)=\sum_{k=0}^{\infty}2^{kz}=\frac{1}{1-2^z},$$

имеем

$$F_4(z)=-\frac{\Gamma(z)}{1-2^z}.$$

Функция $F_4(z)$ имеет двойной полюс в точке $z=0$ и простые полюса в отрицательных целых точках и точках на мнимой оси

$$z_m=\frac{2\pi im}{\ln 2},\quad m=\pm1,\pm2,\ldots.$$

Выберем $\sigma=-1/2$. Найдя вычеты в точках полуплоскости $\Re z>\sigma$, получим

$$f_4(x)=\log_2 x+\frac{\mathcal{C}}{\ln 2}+\frac{1}{2}+\frac{1}{\ln 2}\sum_{m\neq0}\Gamma(z_m)x^{-z_m}+O(x^{-0.5}).$$

Отметим, что гамма-функция убывает в мнимом направлении экспоненциально быстро, поэтому ряд в полученном выражении сходится и является периодической функцией от аргумента $\log_2 x$ с периодом, равным 1.

Ссылки

1. *Диткин В. А., Прудников А. П.* Интегральные преобразования и операционное исчисление. М. : Наука, 1974.

2. *Градштейн И. С., Рыжик И. М.* Таблицы интегралов, сумм, рядов и произведений. М. : Наука, 1971.

УДК 512.711

Н. В. Тимофеева
Ярославский государственный университет им. П. Г. Демидова
E-mail: ntimofeeva@list.ru

Общий случай теоремы об остатках в курсе общей алгебры

Дано легкое диаграммное доказательство обобщенной версии китайской теоремы об остатках для произвольного коммутативного кольца.

Библиография: 3 названия.

Ключевые слова: элементарные гомологические методы, теорема об остатках, коммутативное кольцо, идеал.

Данная заметка посвящена обобщенной версии следующей известной теоремы из теории колец вычетов, включаемой в различные алгебраические и алгоритмические курсы:

Предложение 1 (Китайская теорема об остатках). *Пусть целые числа m и n взаимно просты. Тогда имеет место изоморфизм*

$$\mathbb{Z}_{mn} \cong \mathbb{Z}_m \oplus \mathbb{Z}_n.$$

Будет дано легкое доказательство обобщенной теоремы в рамках развиваемого автором диаграммного подхода [1, 2] в преподавании алгебры студентам-математикам. Рассматриваемая обобщенная версия предложения 1 органично вписывается в программу курса общей алгебры; формулировка теоремы приведена в упражнениях книги Дж. Гэллиана [3, Ch. 3: Rings, Exercise 6, p. 341]:

Теорема. *Пусть R – коммутативное кольцо, I, J – два собственных идеала в R, причем $I + J = R$. Тогда имеет место изоморфизм факторколец*

$$R/(I \cap J) \cong R/I \oplus R/J.$$

Тогда китайская теорема об остатках (предложение 1) является частным случаем приведенной теоремы, как, например, и следующее предложение, приведенное в той же книге [3, Ch. 3: Rings, Exercise 5, p. 341]:

Предложение 2. *Пусть $f(x)$, $g(x)$ – неприводимые полиномы над полем F. Если $f(x)$ и $g(x)$ не ассоциированы в кольце $F[x]$, то имеет место изоморфизм колец*

$$F[x]/ < f(x)g(x) > \cong F[x]/ < f(x) > \oplus F[x]/ < g(x) > .$$

Здесь использовано очевидное обозначение: запись $< f, g >$ означает идеал, порожденный элементами f и g.

Доказательство теоремы получим, рассмотрев точную диаграмму:

$$\begin{array}{ccccccccc}
 & & 0 & & 0 & & 0 & & \\
 & & \uparrow & & \uparrow & & \uparrow & & \\
0 & \longrightarrow & I+J & \longrightarrow & R & \longrightarrow & R/(I+J) & \longrightarrow & 0 \\
 & & \uparrow & & \uparrow & & \uparrow & & \\
0 & \longrightarrow & I\oplus J & \longrightarrow & R\oplus R & \longrightarrow & R/I\oplus R/J & \longrightarrow & 0 \\
 & & \uparrow & & \uparrow & & \uparrow & & \\
0 & \longrightarrow & I\cap J & \longrightarrow & R & \longrightarrow & R/(I\cap J) & \longrightarrow & 0 \\
 & & \uparrow & & \uparrow & & \uparrow & & \\
 & & 0 & & 0 & & 0 & &
\end{array} \qquad (1)$$

Строки диаграммы стандартные. Нижний гомоморфизм среднего столбца определяется соответствием $a \mapsto (a, -a)$, верхний – сложением компонент прямой суммы $(a, b) \mapsto a+b$. Гомоморфизмы левого и правого столбцов индуцированы гомоморфизмами среднего столбца. Применяя условие теоремы, получаем $I + J = R$ и обращение в нуль правого верхнего объекта $R/(I+J) = 0$, откуда следует, что нижняя стрелка правого столбца – изоморфизм.

Замечание. Диаграмма (1) получается для любого коммутативного кольца и любых двух его собственных идеалов. Приведенное доказательство также показывает (не требуя дополнительных действий), что импликация в теореме допускает обращение!

Ссылки

1. *Тимофеева Н. В.* Элементарные гомологические методы в курсах алгебры для математического факультета // Актуальные проблемы совершенствования высшего профессионального образования: материалы XI Областной научно-метод. конф., посвященной 300-летию со дня рождения М. В. Ломоносова. Ярославль : ЯрГУ, 2011. С. 104 – 105.

2. *Тимофеева Н. В.* Диаграммная техника в курсах алгебры для студентов-математиков // Преподавание математики и компьютерных наук в классич. ун-те: материалы 4-й научно-метод. конф. препод. матем. ф-та и ф-та ИВТ Яросл. гос. ун-та им. П. Г. Демидова. Ярославль : ЯрГУ, 2012. С. 73 – 78.

3. *Gallian J. A.* Contemporary Abstract Algebra. Boston : Brooks/Cole: Cengage Learning, 2010.

УДК 519.21

Н. Б. Чаплыгина
Ярославский государственный университет им. П. Г. Демидова
E-mail: chaplgn@uniyar.ac.ru

Задачи на условную вероятность

В статье рассматриваются методические вопросы изложения понятия условной вероятности с примерами задач.

Библиография: 5 названий.

Ключевые слова: вероятностное пространство, условная вероятность, событие, условие.

Условная вероятность – одно из ключевых понятий теории вероятностей. Задачи на условную вероятность традиционно вызывают у студентов заметные трудности на этапе постановки задачи. Приведем в качестве примера следующую задачу.

Пример 1. Три стрелка квалификаций p_1, p_2 и p_3 соответственно стреляют в одну мишень независимо друг от друга. Какова вероятность промаха второго стрелка, если в мишени оказались две пробоины. Предполагается, что разные выстрелы, поразившие мишень, не могут оставить только одну пробоину.

В большой части студенческих решений приводятся следующие рассуждения. Обозначим события попаданий 1-го, 2-го и 3-го стрелков в мишень соответственно через A, B, C. Тогда требуется найти вероятность события $A\overline{B}C$, означающего два попадания и промах второго.

Таким образом, вопрос задачи интерпретируется неверно, связка «если» заменяется связкой «и». Вопрос же задачи заключается в нахождении вероятности события $\overline{B}$, при условии свершившегося события двух попаданий $\overline{A}BC + A\overline{B}C + AB\overline{C}$, т. е. в нахождении условной вероятности.

Условная вероятность является вероятностью апостериорной, имеющей место после наблюдения некоторых событий, после поступления некоторой дополнительной уточняющей информации о складывающейся ситуации.

Условная вероятность $P(A \mid B)$ события A при условии события B определяется как отношение (см. [1],[2])

$$P(A \mid B) = \frac{P(AB)}{P(B)}. \tag{1}$$

За этим формальным определением скрывается, видимо, не совсем очевидный смысл, который становится понятным после применения этой формулы в случае классической схемы – равновозможности элементарных событий конечного вероятностного пространства Ω. Если $N(A)$ – число элементарных исходов события A, то $P(AB) = \frac{N(AB)}{N(\Omega)}$, $P(B) = \frac{N(B)}{N(\Omega)}$. Подставляя эти выражения в формулу (1) условной вероятности, получаем

$$P(A \mid B) = \frac{N(AB)}{N(B)}. \tag{2}$$

Последняя формула имеет понятный интуитивный смысл, но, видимо, она как следствие формулы (1) проходит мимо внимания студентов. Чтобы обратить на нее большее внимание, можно начать изложение условной вероятности именно с классического вероятностного пространства, как это представлено в [5]. Здесь переход к формуле условной вероятности производится как обобщение простейшего примера классического вероятностного пространства. Понятие классической вероятности вызывает у студентов меньше вопросов. В классической схеме определение вероятности $P(A) = \frac{N(A)}{N(\Omega)}$. Если случилось событие B, то можно перейти к новой ситуации – другому множеству элементарных событий Ω', в котором остаются лишь элементарные события множества B, т. е. $\Omega' = B$. В этих условиях событие A представимо лишь элементарными событиями пересечения AB. Тогда нахождение условной вероятности $P(A \mid B)$ сводится к нахождению классической вероятности $P'(A)$ в новом вероятностном пространстве со множеством элементарных событий Ω': $P'(A) = \frac{N(AB)}{N(\Omega')} = \frac{N(AB)}{N(B)}$. Это и есть условная вероятность относительно события B в первоначальном вероятностном пространстве со множеством элементарных событий Ω, т. е. $P(A \mid B)$.

Итак, если в формулировке вопроса задачи на нахождение вероятности события A говорится о том, что известно о наступлении некоторого события B, это означает, что требуется найти условную вероятность $P(A \mid B)$. Наступление события B может быть описано условным предложением со связками: «если», «когда», «при условии», «в предположении», «предполагая, что», аналогичными им и описанием наступления случайного события. Оно может быть описано и отдельным предложением.

Пример 2. Известно, что при бросании 10 костей появилась, по крайней мере, одна единица. Найти вероятность того, что появилось две или более единиц [3, с. 146].

Приведем пример задачи, в которой не требуется применять условную вероятность, несмотря на присутствие связки "если".

Пример 3. Точка случайно бросается на прямоугольник. Какова вероятность, что она окажется ближе к какой-либо диагонали прямоугольника, чем к его сторонам, если длина прямоугольника вдвое больше его ширины?

В вопросе задачи присутствует условное предложение со связкой «если», но нет описания наступления случайного события. Поэтому здесь речь не идет о нахождении условной вероятности. Условное предложение описывает первоначальную ситуацию, продолжает описывать вероятностное пространство.

Рассмотрим еще одну задачу с непростым описанием постановки [4, § 4].

Пример 4. Обрабатываемые на станке детали сортируются на две группы. Каждая очередная деталь независимо от предыдущих с равными вероятностями попадает в первую или вторую группу. Пусть вначале смены для каждой группы деталей приготовлено по ящику емкостью r деталей. Какова вероятность того, что в момент, когда очередную деталь будет некуда класть, в другом ящике будет m деталей?

Разберем ситуацию, описываемую в задаче. «Момент, когда очередную деталь будет некуда класть» описывает попытку поместить очередную деталь в уже полностью укомплектованный деталями ящик. В вопросе о нахождении вероятности присутствует связка «когда» и далее описывается, на первый взгляд, случайное событие. Какое вероятностное пространство можно построить при решении этой задачи? Детали сортируются по очереди до заполнения очередного ящика. Если пронумеровать ящики номерами 1 и 2, то элементарным событием может быть последовательность из номеров 1 и 2, состоящая из $r + 1$ единицы и не более r двоек, заканчивающаяся на 1, а также симметричные последовательности, получающиеся взаимной заменой 1 и 2. Таким образом, случайное событие в вопросе задачи обязательно наступает и оказывается неслучайным. Поэтому речь идет о вероятности безусловной. Такая же по сути задача приводится и в [3, с. 146]. Но постановка задачи содержит подсказки или более понятное описание ситуации.

Пример 5. Задача Банаха о спичечных коробках. Некий математик всегда носит с собой две коробки по N спичек; каждый раз, когда он хочет достать спичку, он выбирает наугад одну из коробок. Неизбежно наступит момент, когда он вынет пустую коробку. В этот момент в другой коробке может быть $r = 0, 1, 2, \ldots, N$ спичек. Задача состоит в том, чтобы найти соответствующие вероятности p_r.

Здесь в описании задачи говорится о неизбежности момента обращения к пустой коробке, что указывает на нахождение безусловной вероятности. Однако можно привлечь к решению задачи и условную вероятность (достоверные события также являются случайными событиями). Пусть $A_{r,1}$ — событие, при котором в первой коробке r спичек, B_2 — со-

бытие обращения ко второй коробке, оказавшейся пустой. Тогда найдем условную вероятность

$$P(A_{r,1} \mid B_2) = \frac{P(A_{r,1}B_2)}{P(B2)}.$$

В описанном в предыдущем примере вероятностном пространстве вероятность $P(B_2) = 1$. Но введение обозначений событий $A_{r,1}$ и B_2 позволяет легче представить исследуемое событие. Искомая вероятность в силу симметрии ситуаций (нумерации коробок) вдвое больше, чем $P(A_{r,1}B_2)$. Аргумент $A_{r,1}B_2$ обозначает событие, заключающееся в том, что произведено $N + (N - r) + 1$ обращений к спичечным коробкам, причем $N + 1$ к одной из них, и последнее в том числе, и $(N - r)$ – к другой. Имеем схему Бернулли с параметром $\frac{1}{2}$ и числом повторений $N + (N - r)$ (без последнего обращения). Искомая вероятность представляется выражением

$$2 \cdot P(A_{r,1}B_2) = 2 \cdot C_{2N-r}^{N} \left(\frac{1}{2}\right)^{2N-r} \cdot \frac{1}{2} = C_{2N-r}^{N} \left(\frac{1}{2}\right)^{2N-r}.$$

Следует избегать постановок задач в такой форме, которая не позволяет однозначно понять итоговый вопрос. В качестве примера приведем вопрос: какова вероятность, что в схеме Бернулли из N испытаний выпало k успехов при том, что первое и последнее испытания закончились успехом? Его можно трактовать двояко. Речевой оборот «притом, что» можно заменить либо связкой «и», либо связкой «если». В этих случаях нужно находить различные величины. В первом – безусловную вероятность произведения событий, во втором – условную (можно предложить решить обе задачи).

Ссылки

1. *Прохоров Ю. В. Розанов Ю. А.* Теория вероятностей. М. : Наука, 1973.

2. *Ширяев А. Н.* Вероятность. М. : Наука, 1980.

3. *Феллер В.* Введение в теорию вероятностей и ее приложения. Т. 1. М. : Мир, 1964.

4. *Севастьянов Б. А. Чистяков В. П. Зубков А. М* Сборник задач по теории вероятностей. М. : Наука, 1980.

5. *Боровков А. А.* Теория вероятностей. М. : Наука, 1976.

УДК 512+519.6

С. И. Яблокова
Ярославский государственный университет им. П. Г. Демидова
E-mail: yabl@uniyar.ac.ru

Китайская теорема об остатках и многомодульная арифметика

Рассматривается вопрос о связи древней теоремы «об остатках» с современными вычислительными методами, в частности с методами, используемыми при работе с очень большими целыми числами, когда требуются точные вычисления.

Библиография : 4 названия.

Ключевые слова: теорема об остатках, кольцо, сравнение, стандатрный набор остатков, вектор основания.

В современной математической литературе эту теорему называют «китайской теоремой об остатках», так как она была известна еще в древнем Китае, с чем связано ее название.

Китайскую математику нельзя рассматривать как изолированную. Начиная с эпохи династии Хань, существовавшей примерно в одно время с Римской империей, Китай поддерживал торговые и культурные связи с другими азиатскими государствами и даже с Европой. На науку Китая влияли также арабская и индийская науки, и такое влияние могло быть взаимным. Так, влияние Индии на Китай могло быть обусловлено проникновением в Китай буддизма в I столетии н. э. Греческое же влияние мало заметно, хотя имеется много сходного в развитии науки, в частности математики.

Во времена династии Тан (618 – 907 гг.) в Китае было изобретено книгопечатание, а в 1115 г. появилось печатное издание «Девяти книг», содержавшее собрание так называемых математических текстов и широко использовавшееся в Китае в течение нескольких столетий.

Но период расцвета древнекитайской математики наступил во времена династии Сун (960 – 1279 гг.). Китайскую теорему об остатках приписывают Сунь-Цзы, жившему по некоторым источникам в I веке н. э., а по другим – в III веке н. э. В своей книге «Суань - Цзинь»

(«Математический трактат») он дает правило «тья-тен» (большое обобщение) определения натурального числа, дающего остатки 2,3,2 при делении на 3,5,7 соответственно. Из числа ведущих математиков Китая того времени стоит также назвать Цинь Цзю-шао, который развивал к тому уже времени давнюю теорию неопределенных уравнений. Один из примеров в его книге, датированной 1247 годом, можно записать следующим образом:

$$x \equiv 32\ (mod\ 83) \equiv 70\ (mod\ 110) \equiv 30\ (mod\ 135).$$

Греческий математик и пифагорейский философ Никомах из Герасы (около 100 г. н. э.) был одним из самых ранних александрийских математиков римского периода. Его труд «Nicomachi Geraseni Pythagorei. Introductionis Arithmeticae» («Арифметическое введение») – наиболее полное из сохранившихся изложений пифагорейской арифметики. Именно в этом сочинении он вводит как игру метод для определения натурального числа по остаткам, полученным от деления этого числа на другие натуральные числа.[1]

Методы решения подобных задач у Никомаха и Сунь-Цзы сходны. Поэтому нельзя однозначно отдать пальму первенства в этом вопросе китайской или греческой математическим школам, но то, что этот метод очень древний, не вызывает сомнений.

В классический алгебраический курс эта теорема не входит, но курс теории чисел уже немыслим без этой теоремы. Алгебраические курсы для будущих специалистов по компьютерной безопасности также не могут обойтись без этой теоремы. Дело в том, что эта древняя теорема лежит в основе многих современных вычислительных методов, использующихся в алгоритмах цифровой обработки сигналов, машинной обработке больших целых чисел, решения задачи интерполяции многочленов над конечным полем и т. д. В курсе «Алгебраическая алгоритмика», читаемом студентам специальности «Компьютерная безопасность» в 4 и 5-м семестрах, эта теорема дается в четырех формулировках и нескольких следствиях, обобщающих простейшее утверждение (теорему 1).

Теорема 1.[2] *Пусть m и n – взаимно простые натуральные числа. Тогда абелева группа $\mathbb{Z}_{mn}$ изоморфна прямому произведению абелевых групп $\mathbb{Z}_m$ и $\mathbb{Z}_n$, т. е.*

$$\mathbb{Z}_{mn} \cong \mathbb{Z}_m \times \mathbb{Z}_n.$$

Следствие. *Если m и n – два взаимно простых натуральных числа, то изоморфизм группы $\mathbb{Z}_{mn}$ в группу $\mathbb{Z}_m \times \mathbb{Z}_n$, ставящий в соответствие элементу $x \in \mathbb{Z}_{mn}$ пару $(x_1, x_2) \in \mathbb{Z}_m \times \mathbb{Z}_n$, где*

$x_1 \equiv x \ (mod\ m)$, $x_2 \equiv x \ (mod\ n)$, *в действительности является изоморфизмом колец.*

Теорема 2.[3] *Пусть* $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_s^{\alpha_s}$, *где* $p_1, p_2, \dots p_s$ *– попарно взаимно простые целые положительные числа,* $\alpha_i > 0$ $(i = 1, \dots, s)$ *– целые. Тогда кольцо* $\mathbb{Z}_m$ *изоморфно прямому произведению колец* $\mathbb{Z}_{p_i^{\alpha_i}}$ $(i = 1, 2, \dots, s)$, *т.е.*

$$\mathbb{Z}_m \cong \mathbb{Z}_{p_1^{\alpha_1}} \times \mathbb{Z}_{p_2^{\alpha_2}} \times \dots \times \mathbb{Z}_{p_s^{\alpha_s}}.$$

Следствие. *Разложение колец, полученное в теореме 2, индуцирует разложение групп их обратимых элементов (мультипликативных групп этих колец)*

$$\mathbb{Z}_m^* \cong \mathbb{Z}_{p_1^{\alpha_1}}^* \times \mathbb{Z}_{p_2^{\alpha_2}}^* \times \dots \times \mathbb{Z}_{p_s^{\alpha_s}}^*.$$

Теорема 3.[3] *Пусть* $m_1, m_2, \dots m_s$ *– попарно взаимно простые целые числа,* $m_i > 1$ $(i = 1, 2, \dots, s)$, *и пусть* $M = m_1 m_2 \dots m_s$. *Тогда существует единственное неотрицательное решение по модулю* M *следующей системы сравнений:*

$$\begin{cases} x \equiv a_1 \ (mod\ m_1) \\ x \equiv a_2 \ (mod\ m_2) \\ \dots\dots\dots\dots\dots \\ x \equiv a_s \ (mod\ m_s). \end{cases} \qquad (1)$$

Теорема 4.[4] *Пусть* $m_1, m_2, \dots m_s$ *– попарно взаимно простые целые числа,* $m_i > 1$ $(i = 1, 2, \dots, s)$, *и пусть* $M = m_1 m_2 \dots m_s$. *Тогда единственным неотрицательным решением по модулю* M *системы сравнений (1) является*

$$x \equiv \sum_{i=1}^{s} a_i M_i N_i \ \ (mod\ M),$$

где $M_i = \dfrac{M}{m_i}$, $i = 1, 2, \dots, s$, *а* N_i *– целое, удовлетворяющее условию*

$$M_i N_i + m_i n_i = 1 \qquad (i = 1, 2, \dots, s).$$

Доказательство теоремы 1 конструктивно и дает метод решения системы из двух сравнений, который легко обобщить на систему из s сравнений по взаимно простым модулям. Это обобщение приводит к представлению решения системы из s сравнений в виде

$$x = q_0 + m_1 q_1 + m_1 m_2 q_2 + \dots + m_1 m_2 \dots m_{s-1} q_{s-1}. \qquad (2)$$

Это так называемое греко-китайское представление целого числа x. Именно, произвольное целое число x, $|x| < M$, где $M = m_1 m_2 \dots m_s$ и

НОД$(m_i\,,\,m_j)=1$ для $i\neq j$, однозначно представимо своими наименьшими неотрицательными остатками по модулям m_i $(i=1,2,\ldots,s)$.

Вычисления, использующие многомодульную арифметику, основаны на китайской теореме об остатках. Такие вычисления представляют собой способ выполнения точных арифметических действий с большими целыми числами. В основе этого метода лежит изоморфизм кольца $\mathbb{Z}_M$ и прямого произведения колец $\mathbb{Z}_{m_i}$ $(i=1,2,\ldots,s)$.

Пусть требуется вычислить значение многочлена $f(x_1,x_2,\ldots,x_n)$, зависящего от целочисленных аргументов $x_1,\ x_2,\ \ldots,x_n$. Допустим, что можно заранее оценить величину $r=f(x_1,x_2,\ldots,x_n)\in\mathbb{Z}$. Взаимно простые небольшие модули выбираются так, чтобы $m_1m_2\ldots m_s>|r|$. Вектор оснований многомодульной арифметики состоит из упорядоченного набора взаимно простых чисел m_i $(i=1,\ldots,s):\ \beta=\{m_1,\ m_2,\ \ldots,\ m_s\}$. Каждому целому числу x ставится в соответствие стандартный набор остатков числа x относительно данного вектора оснований β :

$$x\ (mod\ \beta)=\{x_1,\ x_2,\ \ldots,\ x_s\},$$

где $x_j\equiv x\ (mod\ m_j),\ \ 0\leq x_j<m_j\ (i=1,2,\ldots,s)$.

Из китайской теоремы об остатках вытекает справедливость следующего утверждения.

Теорема. *Два целых числа x и y имеют одинаковые стандартные наборы остатков относительно вектора оснований $\beta=\{m_1,\ m_2,\ \ldots,\ m_s\}$ тогда и только тогда, когда $x\equiv y\ (mod\ M)$, $M=m_1m_2\ldots m_s$.*

Отсюда следует, что многомодульная арифметика эквивалентна арифметике в кольце $\mathbb{Z}_M$. Главное преимущество такой числовой системы состоит в отсутствии переносов при выполнении операций сложения и умножения. Арифметические действия выполняются полностью и независимо в разных позициях, поэтому выполнять сложение и умножение длинных (больших) целых чисел можно так же быстро, как и обычных (коротких) чисел. Представление (2) здесь также используется для того, чтобы определить знак числа, не вычисляя само это число. Для этой цели требуется найти коэффициенты в представлении (2). Несложный алгоритм позволяет сделать это, используя арифметические действия в многомодульном представлении.[4]

В дальнейшем та же многомодульная идея возникает в разделе курса, посвященного быстрым алгоритмам, а также при обсуждении некоторых разделов курса «Теоретико-числовые методы в криптографии».

Ссылки

1. *Стройк Д.Я.* Краткий очерк истории математики. М. : Наука, 1984.

2. *Ноден П., Китте К.* Алгебраическая алгоритмика. М. : Мир, 1999.

3. *Акритас А.* Основы компьютерной алгебры с приложениями. М. : Мир, 1994.

4. *Яблокова С. И.* Основы алгебраической алгоритмики. Ч. 1 : учебное пособие. Ярославль : ЯрГУ, 2008.

УДК 512+519.725

С. И. Яблокова
Ярославский государственный университет им. П. Г. Демидова
E-mail: yabl@uniyar.ac.ru

Китайская теорема об остатках, модульная арифметика и быстрые алгоритмы цифровой обработки сигналов

Рассматривается связь между «китайской теоремой об остатках», эффективными вычислениями в многомодульной арифметике и быстрыми алгоритмами цифровой обработки сигналов.

Библиография : 4 названия.

Ключевые слова: теорема об остатках, дискретное преобразование Фурье, быстрые алгоритмы.

В связи с использованием многомодульной арифметики для точных вычислений с очень большими целыми числами возникает и чисто алгоритмический вопрос, связанный со сложностью (или трудоемкостью) вычислений, а именно: как с наименьшей трудоемкостью перейти от числа x к стандартному набору остатков относительно данного вектора оснований? Часто используемый метод, который здесь применяют, не только пригоден для многомодульной арифметики, но как мы увидим ниже, приводит к идее быстрого алгоритма вычисления дискретного преобразования Фурье.

Этот метод в литературе иногда называют методом «разделяй и властвуй». Предположим (для простоты рассуждений), что s – количество малых модулей m_i, используемых в многомодульной арифметике, равно степени двойки. Тогда вместо того, чтобы для каждого i вычислять вычет $x \ (mod \ m_i)$, предлагается проделать следующую последовательность вычислений.

Вычислять последовательно числа:
сначала $\frac{s}{2}$ чисел $m_i^{(1)} = m_{2i-1}m_{2i} \quad \left(i = 1, 2, \ldots, \frac{s}{2}\right)$, затем $\frac{s}{2^2}$

чисел $m_i^{(2)} = m_{2i-1}^{(1)} m_{2i}^{(1)} \left(i = 1, 2, \ldots, \frac{s}{2^2}\right)$ и т.д. до $m_1^{(k-1)} = m_1^{(k-2)} m_2^{(k-2)}$, $m_2^{(k-1)} = m_3^{(k-2)} m_4^{(k-2)}$ $(k = log_2\ s)$.

Теперь, имея эти произведения, последовательно вычислять промежуточные вычеты числа x. Сначала вычислить $x_{11} \equiv x\ (mod\ m_1^{(k-1)})$ и $x_{12} \equiv x\ (mod\, m_2^{(k-1)})$, затем для каждого из полученных вычетов $x_{11},\ x_{12}$ вычислить пару вычетов $x_{21} \equiv x_{11}\ (mod\, m_1^{(k-2)})$, $x_{22} \equiv x_{11}\ (mod\, m_2^{(k-2)})$, $x_{23} \equiv x_{12}\ (mod\ m_3^{(k-2)})$, $x_{24} \equiv x_{12}\ (mod\ m_4^{(k-2)})$. Далее, на каждом шаге для каждого из полученных вычетов $x_{jl}\,(\equiv x\ (mod\ m_l^{(k-j)}))$ вычислять два вычета по модулям двух сомножителей числа $m_l^{(k-j)}$, а именно, $m_{2l-1}^{(k-j-1)}$ и $m_{2l}^{(k-j-1)}$ и так до тех пор, пока не получатся вычеты x по модулям $m_1,\ m_2,\ \ldots, m_s$.

Такая на первый взгляд длинная процедура вычисления стандартного набора остатков на самом деле в пересчете на побитовые операции выполняется быстрее, чем прямое приведение x по модулям $m_i\ (i = 1, 2, \ldots, s)$.

Та же идея лежит в основе быстрого алгоритма Кули-Тьюки по основанию 2 для вычисления дискретного преобразования Фурье[1,2]. Дело в том, что на вычисление дискретного преобразования Фурье вектора $\mathbf{v} = \{v_0,\ v_1,\ \ldots,\ v_{n-1}\}$ с n компонентами, т. е. на получения компонент нового вектора $\mathbf{V} = \{V_0,\ V_1,\ \ldots,\ V_{n-1}\}$

$$V_k = \sum_{j=0}^{n-1} \omega^{jk}\, v_j \qquad (k = 0, 1, \ldots, n-1)$$

можно смотреть как на вычисление значений многочлена

$$f(x) = \sum_{j=0}^{n-1} v_j\, x^j$$

в точках $\omega^k\ \ (k = 0, 1, \ldots, n-1)$, т. е.

$$V_k \equiv f(x)\ (mod\ x - \omega^k) \qquad (k = 0, 1, \ldots, n-1).$$

Так как ω – корень n-й степени из 1, то множество $\{\omega^k \mid k = 0, 1, \ldots, n-1\}$ дает нам все корни степени n из 1. Т. е. числа ω^k являются корнями многочлена $x^n - 1$. Поэтому, чтобы найти компоненты V_k, надо найти вычеты многочлена $f(x)$ по модулям $x - \omega^k$, являющимся сомножителями многочлена $x^n - 1$.[3]

Если $n = 2^l$, то можно рассматривать следующее разложение многочлена $x^n - 1$ на множители. Сначала рассмотрим разложение

$$x^n - 1 = (x^{\frac{n}{2}} - 1)\,(x^{\frac{n}{2}} - \omega^{\frac{n}{2}})$$

и ищем вычеты $f(x)$ по каждому многочлену $x^{\frac{n}{2}}-1$ и $x^{\frac{n}{2}}-\omega^{\frac{n}{2}}$, т. е. получим два вычета $f_{11}(x)$ и $f_{12}(x)$. Далее, каждый из использованных многочленов-модулей можно опять разложить в произведение двух многочленов

$$x^{\frac{n}{2}}-1=(x^{\frac{n}{4}}-1)\,(x^{\frac{n}{4}}-\omega^{\frac{n}{2}}),$$
$$x^{\frac{n}{2}}-\omega^{\frac{n}{2}}=(x^{\frac{n}{4}}-\omega^{\frac{n}{4}})\,(x^{\frac{n}{4}}-\omega^{\frac{3n}{4}}),$$

вычислить вычеты $f_{11}(x)$ и $f_{12}(x)$ по двум соответствующим модулям и продолжать этот процесс до тех пор, пока не получатся вычеты $f(x)$ $(mod\ x-\omega^k)$, т. е. компоненты вектора $\mathbf{V}$. Приведение многочлена степени $n-1$ по модулю многочлена $x^{\frac{n}{2}}-c$ требует всего $O(n\,log_2\,n)$ арифметических операций и сводится к разбиению исходного вектора $\mathbf{v}$ на две равные половины: первые $\dfrac{n}{2}$ компонент и последние $\dfrac{n}{2}$ компонент, умножению каждой компоненты второй половины на константу c и сложению соответствующих компонент $v_j+cv_{j+\frac{n}{2}}$, $\left(j=0,1,\ldots,\dfrac{n}{2}-1\right)$, поскольку

$$\sum_{j=0}^{n-1} v_j\,x^j\ (mod\ x^{\frac{n}{2}}-c)\equiv\sum_{j=0}^{\frac{n}{2}-1}(v_j+cv_{j+\frac{n}{2}})x^j.$$

Эта простая процедура позволяет значительно уменьшить трудоемкость вычислений, т. е. получить быстрый алгоритм ДПФ.

Прослеживание подобных связей между разделами курса позволяет глубже понять изучаемый предмет, связать между собой многие, на первый взгляд, казалось бы, различные, алгоритмы и методы решения алгоритмических задач.

Возвращаясь к китайской теореме об остатках, следует отметить, что перенесенная из кольца целых чисел в кольцо многочленов над полем, эта теорема находит свое применение при построении быстрых алгоритмов вычисления линейных и циклических сверток. Формулировка и доказательство этой теоремы дается в 5-м семестре курса алгебраической алгоритмики при изучении алгоритмов в кольце многочленов.

Теорема.[4] *Для заданного множества попарно взаимно простых многочленов $m_1(x)$, $m_2(x)$, …, $m_s(x)$ из кольца $K[x]$ (K – поле) и множества многочленов $c_i(x)$ $(i=1,2,\ldots,s)$, $c_i(x)\in K[x]$ таких, что $deg\ c_i(x)<deg\ m_i(x)$ $(i=1,2,\ldots,s)$, система сравнений*

$$f(x)\equiv c_i(x)\ (mod\ m_i(x)),\qquad i=1,2,\ldots,s,$$

имеет не более одного решения $f(x)$, удовлетворяющего условию

$$deg\ f(x)<\sum_{i=1}^{s} deg\ m_i(x),$$

которое можно найти по формуле

$$f(x) = \sum_{i=1}^{s} c_i(x)\, M_i(x)\, N_i(x) \pmod{M(x)},$$

где $M(x) = m_1(x)\, m_2(x) \ldots m_s(x)$, $M_i(x) = \dfrac{M(x)}{m_i(x)}$ $(i = 1, 2, \ldots, s)$, *а* $N_i(x)$ *удовлетворяет условию*

$$M_i(x)\, N_i(x) + m_i(x)\, n_i(x) = 1 \qquad (i = 1, 2, \ldots, s).$$

Задача вычисления линейной свертки двух числовых последовательностей в терминах многочленов сводится к вычислению коэффициентов многочлена $s(x)$, равного произведению двух многочленов $d(x)$ и $g(x)$, коэффициентами которых служат соответственно компоненты двух данных числовых последовательностей. Один из самых эффективных быстрых алгоритмов вычисления сверток в полях $\mathbb{R}$ и $\mathbb{C}$, предложенный в 1976 – 1978 гг. Виноградом, опирается на китайскую теорему об остатках для многочленов. Виноград не только предложил этот метод, он сумел доказать, что этот метод позволяет в каждом случае построить наиболее эффективный по количеству умножений алгоритм вычисления свертки.

Идея состоит в том, чтобы сначала найти вычеты искомого многочлена $s(x)$ по некоторому набору взаимно простых многочленов-модулей, а затем восстановить его, пользуясь китайской теоремой об остатках. При этом многочлены - модули выбираются так, чтобы приведение исходных двух многочленов $d(x)$ и $g(x)$ по этим модулям содержало как можно меньше операций умножения, в наилучшем случае эти многочлены-модули являются унитарными многочленами первой степени с малыми целыми корнями. В таком случае при вычислении линейной свертки последовательностей длин N и L можно обойтись $N+L-1$ умножениями. Поскольку циклическая свертка двух многочленов одинаковой степени $n-1$ получается из линейной свертки этих многочленов приведением результата по модулю многочлена $x^n - 1$, что сводится к вычислению их линейной свертки и некоторым дополнительным сложениям, то быстрые алгоритмы циклических сверток можно строить, исходя из тех же соображений.

Еще одно замечание касается теоремы о свертке доказанной Стогхемом, который вскоре после опубликования алгоритма Кули-Тьюки в 1965 г. заметил, что быстрые алгоритмы вычисления преобразования Фурье могут служить удобным способом вычисления сверток. Это и послужило толчком к развитию нового вычислительного раздела математики, получившего название «быстрые алгоритмы». Будем интерпретировать циклическую свертку двух числовых последовательностей

длины n каждая как последовательность коэффициентов многочлена $\tilde{s}(x) \equiv d(x)\, g(x)\ (mod\ x^n - 1)$. Так как корнями многочлена $x^n - 1$ являются все корни n-й степени из 1, то приходим к тому же методу вычисления, как в методе «разделяй и властвуй». Можно с помощью этого метода вычислить значения многочленов $d(x)$ и $g(x)$ в каждой точке $\omega^k \quad (k = 0, 1, \ldots, n-1)$:

$$d(\omega^k) \equiv d(x)\ (mod\ x - \omega^k)$$
$$g(\omega^k) \equiv g(x)\ (mod\ x - \omega^k),'$$

а тогда $\tilde{s}(\omega^k) = d(\omega^k)\, g(\omega^k), \quad (k = 0, 1, \ldots, n-1)$.

Далее, по известным значениям многочлена $\tilde{s}(x)$ в точках ω^k остается построить этот многочлен. Получаем задачу интерполяции, которая сводится к обратному преобразованию Фурье, т. е. вычислению компонент вектора $\mathbf{v}$ по известному преобразованию Фурье $\mathbf{V}$ этого вектора

$$v_j = \frac{1}{n} \sum_{k=0}^{n-1} \omega^{-jk}\, V_k, \qquad j = 0, 1, \ldots, n-1.$$

Для вычисления этого преобразования, очевидно, опять можно применить тот же метод.

Благодаря алгоритму БПФ, дискретное преобразование Фурье является удобным инструментом при проведении вычислений с многочленами. С его помощью можно вычислить произведение двух многочленов со сложностью $O(n\, log_2\, n)$. А это, в свою очередь, приводит к идее алгоритма быстрого битового умножения больших целых чисел, который известен как алгоритм Шёнхаге-Штрассена и является весьма нетривиальным.

Ссылки

1. *Ноден П., Китте К.* Алгебраическая алгоритмика. М. : Мир, 1999.

2. *Блейхут Р. Э.* Быстрые алгоритмы цифровой обработки сигналов. М. : Мир, 1979.

3. *Яблокова С.И.* Введение в быстрые алгоритмы цифровой обработки сигналов : учебное пособие. Ярославль : ЯрГУ, 2009.

4. *Яблокова С. И.* Основы алгебраической алгоритмики. Ч. 2 : учебное пособие. Ярославль : ЯрГУ, 2009.

УДК 519.683.8

О. П. Якимова
Ярославский государственный университет им. П. Г. Демидова
E-mail: polya@uniyar.ac.ru

Создание многопоточных приложений в .Net 4.0

В докладе приводится краткий обзор средств, представляемых средой .Net Framework 4.0, для создания многопоточных приложений.

Библиография: 1 название.

Ключевые слова: многопоточность, параллельная обработка данных, C#, .Net.

В настоящее время многие персональные компьютеры и рабочие станции имеют центральные процессоры, содержащие два или четыре ядра, которые позволяют одновременно выполнять несколько потоков. В ближайшем будущем ожидается, что компьютеры будут иметь значительно больше ядер. Поэтому актуальной является задача распараллеливания кода, чтобы полностью использовать новые возможности оборудования и повысить производительность разрабатываемого программного обеспечения. В рамках дисциплины «Языки программирования» в 4-м семестре для студентов специальности Компьютерная безопасность излагается материал по созданию многопоточных приложений и проводится соответствующая лабораторная работа.

Платформа .Net достаточно давно предоставляла средства для создания многопоточных приложений на основе асинхронного вызова методов с помощью делегатов, а также использования классов Thread и ThreadPool, но эти инструменты требовали управления потоками и взаимоблокировками на низком уровне.

В версии 4.0 среды .NET Framework появляется библиотека распараллеливания задач (TPL). Эта библиотека усовершенствует многопоточное программирование двумя основными способами. Во-первых, она упрощает создание и применение многих потоков. И во-вторых, она позволяет автоматически использовать несколько процессоров. Библиотека TPL занимается распределением работы, планированием потоков,

управлением состоянием и прочими низкоуровневыми деталями. В результате появляется возможность максимизировать производительность приложений .NET, не имея дела со сложностями прямой работы с потоками.

Фактически параллелизм в программу можно ввести двумя основными способами. Первый из них называется параллелизмом данных. При таком подходе одна операция над совокупностью данных разбивается на два параллельно выполняемых потока или больше, в каждом из которых обрабатывается часть данных. Так, для изменения каждого элемента некоторого контейнера данных большого объема можно организовать параллельную обработку разных частей массива в двух или больше потоках. Нетрудно догадаться, что такие параллельно выполняющиеся действия могут привести к значительному ускорению обработки данных по сравнению с последовательным подходом. Второй способ ввода параллелизма называется параллелизмом задач. При таком подходе две операции или больше выполняются параллельно. Приведем примеры, демонстрирующие оба подхода.

Параллельная обработка данных

Одним из главных классов в TPL является класс $Parallel$. Он поддерживает набор методов, которые позволяют выполнять итерации по коллекции данных, реализующей $IEnumerable < T >$ в параллельном режиме. А именно, этот класс поддерживает два статических метода — $Parallel.For()$ и $Parallel.ForEach()$, которые позволяют выполнять циклическую обработку данных в параллельном режиме. Но нужно будет использовать делегаты $System.Func < T >$ и $System.Action < T >$ для указания целевого метода, который будет вызываться для обработки данных ($Action < T >$ принимает метод, который может возвращать только $void$, $Func < T >$ представляет метод, который возвращает значение и принимает различное количество параметров).

Формальное описание метода For:

$$public\ static\ ParallelLoopResult$$
$$For(int\ fromInclusive, int\ toExclusive, Action < int >\ body),$$

где $fromInclusive$ обозначает начальное значение переменной управления циклом, $toExclusive$ — значение, на единицу больше конечного. На каждом шаге цикла переменная управления циклом увеличивается на единицу. Циклически выполняемый код указывается методом, передаваемым через параметр $body$. Этот метод должен быть совместим с делегатом $Action < int >$, причем может задаваться анонимно или через лямбда-выражение. Метод $For()$ возвращает экземпляр объекта типа $ParallelLoopResult$, описывающий состояние завершения цикла, но этим значением можно пренебречь для простых циклов.

Метод *For*() позволяет, когда такая возможность имеется, распараллелить исполнение кода в цикле, что часто ведет к повышению производительности при обработке больших массивов данных. Но следует учесть, что на распараллеливание цикла также уходит время и поэтому для коротких циклов или небольших массивов этот метод невыгоден.

Ниже приведен пример программы, где создается массив из десяти миллионов целых случайных чисел и его копия. Над элементами массива производятся некоторые преобразования двумя способами – последовательной итерацией и с помощью параллельных вычислений, а также замеряется время работы.

```
class Program
{
    public static void Main()
    {
    // создаем большой массив случайных целых чисел
        const int arLen = 10000000;
        int[] largeArInts = new int[arLen];
        Random r = new Random();
        for (int i = 0; i < largeArInts.Length; i++)
        {
            largeArInts[i] = r.Next(-500000, 500000);
        }
        var copyArInts = new int[arLen];
        Array.Copy(largeArInts, copyArInts, arLen);
        Stopwatch tS = new Stopwatch();
        tS.Start();
        LineArrayProcessing(largeArInts);
        tS.Stop();
        Console.WriteLine(tS.ElapsedMilliseconds);
        tS.Reset();
        tS.Start();
        ParrallelArrayProcessing(copyArInts);
        tS.Stop();
        Console.WriteLine(tS.ElapsedMilliseconds);
     }

    static void LineArrayProcessing(int[] ints)
    {
        for (int i = 0; i < ints.Length; i++)
        {
            if (ints[i] < 0)  ints[i] = -100;
            if (ints[i] >= 100000) ints[i] = 1000;
            if (ints[i] > 200000) ints[i] = 2000;
```

```
            if (ints[i] > 300000) ints[i] = 3000;
            if (ints[i] > 400000) ints[i] = 4000;
        }
    }

    static void ParrallelArrayProcessing(int[] ints)
    {
        Parallel.For(0, ints.Length, i =>
        {
            if (ints[i] < 0)  ints[i] = -100;
            if (ints[i] >= 100000) ints[i] = 1000;
            if (ints[i] > 200000) ints[i] = 2000;
            if (ints[i] > 300000) ints[i] = 3000;
            if (ints[i] > 400000) ints[i] = 4000;
        });
    }
}
```

Параллельный вариант цикла преобразования данных выполняется быстрее, чем последовательный почти в два раза (время работы 86 мс и 170 мс). Для массива из миллиона элементов время обработки будет одинаковым, а для данных меньшего размера быстрее будет последовательная обработка.

Параллелизм задач

В основу TPL положен класс *Task*, который определен в пространстве имен *System.Threading.Tasks*. Наиболее простым способом создания потока с использованием этого класса является вызов метода *StartNew* у статического свойства *Factory*:

```
Task.Factory.StartNew(CheckPrimeNumber);
```

где *CheckPrimeNumber* – имя метода, совместимого с делегатом *Action*. Свойство *Factory* возвращает объект класса *TaskFactory*. Этот объект отвечает за создание объектов типа *Task*. У метода *StartNew* есть много перегрузок, позволяющих создавать потоки, которые возвращают результаты, потоки, которым передаются параметры, и т. д.

В приведенной ниже программе на практике показываются различные способы создания задач.

```
using System;
using System.Threading.Tasks;

namespace Tasks
{
    class Program
    {
```

```
        public static void Main()
        {
            long num = 1000000021;
            Task.Factory.StartNew(CheckNumber, num);
            Task.Factory.StartNew(SomeMethod);
       // Использование конструктора Task
            Task task1 = new Task(SomeMethod);
             task1.Start();
             Console.ReadKey();
        }
        static void SomeMethod()
        {
             Console.WriteLine("поток запущен");
             for (int count = 0; count < 1000; count++)
             {
                  Console.WriteLine(count*count);
             }
        }
        static void CheckNumber(object state)
        {
            long num = Convert.ToInt64(state);
            for (long d = 2; d < num / 2; d++)
            {
                if (num % d == 0)
                {
                    Console.Write("{0} is not prime", num);
                    Console.WriteLine("Divider is {0}", d);
                      return;
                 }
             }
             Console.WriteLine("{0} is prime", num);
          }
    }
}
```

Класс $Task$ позволяет получить результаты выполнения потока. Для этого используется обобщенная версия этого класса $Task < T >$. Пусть у нас есть метод, возвращающий некоторое значение:

```
private static long GetDivider(object state)
```

И мы хотим исполнить его асинхронно и получить результат. Для этого создается экземпляр класса $Task < long >$:

```
long num = 1000000021;
var tk = Task<long>.Factory.StartNew(GetDivider, num);
```

Затем необходимо любым способом дождаться завершения выполнения задачи и получить результат через свойство *Result*:

```
Console.Write("Divider of {0} is {1}", num, tk.Result);
```

Это очень удобно по двум причинам. Во-первых, это означает, что с помощью задачи можно вычислить некоторый результат. Подобным образом поддерживаются параллельные вычисления. И во-вторых, вызывающий процесс окажется блокированным до тех пор, пока не будет получен результат. Это означает, что для организации ожидания результата не требуется никакой особой синхронизации.

Одной из новаторских и очень удобных особенностей библиотеки TPL является возможность создавать продолжение задачи. Продолжение — это одна задача, которая автоматически начинается после завершения другой задачи. Создать продолжение можно с помощью метода *ContinueWith*(), определенного в классе *Task* или с помощью методов *ContinueWhenAll* и *ContinueWhenAny* класса *TaskFactory*. Приведем небольшой пример кода:

```
    Task task = new Task(CheckPrimeNumber);
task.ContinueWith(GetDividerFinished);
task.Start();
```

Изучение студентами библиотеки TPL открывает возможности для автоматического масштабирования приложений с целью эффективного использования ряда доступных процессоров.

Дополнительную информацию о параллельном программировании и средствах библиотеки TPL можно найти в документации (URL : http://msdn.microsoft.com/ru-ru/library/dd460693(v=vs.110).aspx) и в статье [1].

Ссылки

1. *Речкунов Д.* Параллельное программирование в .NET Framework 4.0. URL : http://www.enterra.ru/blog/parallel-programming-in-net-framework-4-0/

Научное издание

5-я научно-методическая конференция
преподавателей математического факультета
и факультета информатики
и вычислительной техники ЯрГУ

Преподавание математики и компьютерных наук в классическом университете

Материалы конференции

Редактор, корректор М. В. Никулина
Компьютерная верстка А. Ю. Ухалов

Подписано в печать 09.06.2014. Формат 60×84 1/8.
Усл. печ. л. 19,53. Уч.-изд. л. 7,0.
Тираж 70 экз. Заказ

Оригинал-макет подготовлен
в редакционно-издательском отделе ЯрГУ.

Ярославский государственный университет им. П. Г. Демидова
150000, г. Ярославль, ул. Советская, 14.