

Крымский научный центр НАН Украины

Таврический национальный университет  
им. В.И. Вернадского

Филиал Московского государственного университета  
им. М. В. Ломоносова в г. Севастополе

**ТАВРИЧЕСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ  
СТУДЕНТОВ И МОЛОДЫХ СПЕЦИАЛИСТОВ  
ПО ИНФОРМАТИКЕ И МАТЕМАТИКЕ**

20-23 апреля 2010

КНЦ НАНУ  
Симферополь

**Редакционный совет:**

**Донской В. И.**, д.ф.-м.н., профессор, гл. научный сотрудник  
КНЦ НАН Украины

**Копачевский Н. Д.**, д.ф.-м.н., профессор

**Чехов В. Н.**, д.ф.-м.н., профессор

**Орлов И. В.**, д.ф.-м.н., профессор

**Марянин Б. Д.**, к.ф.-м.н., доцент, ответственный редактор

**Техническая группа:**

**Старков П. А.**, к.ф.-м.н., доцент

**Войтицкий В. И.**, ассистент, технический редактор

**Литвиненко А. Л.**, студент

**СЕКЦИИ НАУЧНОЙ КОНФЕРЕНЦИИ:**

1. МАТЕМАТИКА, председатель Копачевский Н. Д.  
математический анализ, дифференциальные уравнения, алгебра,  
функциональный анализ, геометрия
2. ПРИКЛАДНАЯ МАТЕМАТИКА, председатель Чехов В. А.  
математические модели в естествознании, численные методы решения  
математических задач
3. ИНФОРМАТИКА, председатель Донской В. И.  
прикладная информатика, программное обеспечение компьютерных систем

Печатается по решению научно-технического Совета КНЦ НАН и  
Министерства образования и науки Украины от 14 апреля 2010 года,  
протокол 4.

подписано к печати 16.04.2010 г.  
отпечатано в типографии КНЦ НАНУ

## Секция МАТЕМАТИКИ

### Компьютерные технологии в преподавании курса стереометрии в школе

*Абибулаева С. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 404)

В последнее десятилетие все большее значение приобретает компьютер, как средство коммуникации, а овладение им хотя бы на уровне пользователя, становится одним из условий эффективного участия в социальной жизни. Особенно необходим компьютер для обучения учащихся и должен занять в данном процессе достойное место. Применение вычислительной техники в учебном процессе открывает новые пути в развитии навыков мышления и умения решать сложные задачи, предоставляет принципиально новые возможности для активизации обучения.

В связи с этим ведутся поиски новых эффективных методов обучения. В 1999 году была создана программа «Стереометрия». Данное издание разработано издательством «КОРДИС & МЕДИА» совместно с «Компьютерным Учебным Детским Информационным Центром» («КУДИЦ»). Представляет собой полный курс стереометрии в объеме средней школы, максимально использующий возможности современного персонального компьютера для удобного и наглядного представления учебной информации.

Издание состоит из «Учебника», «Зачета» и встроенного трехмерного редактора чертежей «Стереоконструктор». «Стереоконструктор» предназначен для решения задач на построение, а также для создания графических и анимационных иллюстраций. В свою очередь, «Учебник» включает в себя: «Теорию», в которой изложены основные сведения по стереометрии в объеме курса геометрии 10-11 классов средней школы (66 тем); дополнительные теоремы, формулы и примеры для углубленного изучения каждой темы; а также представлены более 3000 трехмерных интерактивных иллюстраций, как к основному, так и к дополнительному учебному материалу; «Задачник», в котором содержится подбор типовых задач (более 420), ориентированный на подготовку к выпускным экзаменам в школе и к вступительным экзаменам по математике в ВУЗы. Каждая задача снабжена ссылками на необходимый для ее решения теоретический материал; «Рабочую тетрадь», дающую возможность

дополнить материал издания собственными разработками, задачами и их решениями, обменяться ими с коллегами - преподавателями и методистами - через Интернет. В разделе «Зачет» предлагается набор задач (количество задач определяет пользователь), который случайным образом формируется из задач по выбранному разделу или по всему курсу. Имеется шкала, отображающая количество выбранных задач, количество решенных и количество правильно решенных задач.

Однако не все педагоги на нынешнее время обладают навыками пользования современными техническими средствами обучения. Мои разработки данной программы содержат детальнейшее описание электронного учебника, а также поурочные планы для проведения уроков геометрии в 10-11 классах с использованием диска «Стереометрия». Мною изучены и описаны все разделы «Стереометрии». Описание программы подробное, содержит таблицы, в которых приведена характеристика инструментов со всех панелей редактора чертежей «Стереоконструктор». Рассмотрены с использованием данного электронного учебника такие темы школьного курса геометрии: пересечение прямой и плоскости; создание линии пересечения плоскостей; создание прямой, перпендикулярной плоскости и проходящей через данную точку; создание точки пересечения прямой и плоскости; создание перпендикулярной плоскости; построение высоты, медианы и биссектрисы треугольника; сечение многоугольных пирамид и многоугольных призм. А также разработаны поурочные планы с применением «Стереометрии» по таким темам: «Расположение прямых в пространстве», «Двугранный угол» и «Цилиндр».

Моей задачей является с помощью данных разработок создать методическое пособие, содержащее инструкцию пользования электронным учебником и поурочные планы по геометрии с использованием данной программы. Такое пособие поможет любому ученику ознакомиться с программой, а учителю улучшить методическую подготовленность к использованию компьютера, как педагогического элемента урока математики. В дальнейшем планирую рассмотреть элементы аналитической геометрии при помощи программы «Стереометрия».

## Использование электронных учебников в преподавании курса планиметрии

*Анисимова И. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 404)

Внедрение новых информационных технологий в учебный процесс меняет традиционный взгляд на образование. Мультимедийные компьютерные программы позволяют разгрузить учителя, увеличить заинтересованность учащихся в предмете, более наглядной подачи материала за счет анимации. Использование электронных учебников ориентировано на поддержку традиционного курса обучения, что не отвлекает детей от школьного предмета, а способствует развитию у ребенка повышенного интереса к нему. Это использование осуществляется при участии и под руководством учителя. При решении задач с помощью электронного учебника не тратится время на построение чертежей. Еще одно очень важное преимущество - индивидуализация обучения. Это позволяет выявить и развить личные неповторимые способности каждого.

Одними из таких электронных учебников является «Планиметрия 7-9 класс», издаваемый «КОРДИС & МЕДИА», разработанный Образовательным Центром «КУДИЦ» и пакет динамической геометрии DG (Dynamic Geometry), разработанный в Региональном центре новых информационных технологий при ХГПУ им.Г.С.Сковороды. Данные издания представляют собой полный курс планиметрии в объеме средней школы, использующие возможности современного персонального компьютера для удобного и наглядного представления учебной информации.

«Планиметрия 7-9 класс» содержит в себе «Учебник», «Справочник», «Задачник», «Редактор чертежей», «Зачет». Четко прописаны все разделы курса, каждый из которых делится на темы, в которых отображается количество определений, теорем и задач. Изложены дополнительные теоремы и формулы, что предусматривает углубленное изучение каждой темы. Представлено огромное количество иллюстраций к основному и дополнительному учебному материалу. Все формулировки и доказательства озвучены. Четко представлены доказательства теорем; они выполнены пошагово с помощью управляющих кнопок. Причем все дополнительные построения мгновенно отображаются на чертеже, сопровождаясь звуковым сигналом (при желании его можно выключить). Сами чертежи красочные, аккуратные, со всеми надлежащими обозначениями.

В «Задачнике» содержится хороший подбор как стандартных, так и нестандартных задач (около 500 задач с указаниями к решению). В разделе «Зачет» предлагается набор задач, который случайным образом формируется из задач по выбранному разделу или по всему курсу. Проверка правильности решения отображается индикатором после введения соответствующего ответа. «Редактор чертежей» предназначен для решения задач на построение, а также для создания графических и анимационных иллюстраций.

Целью DG является предоставление ученикам возможности самостоятельного открытия геометрии путем экспериментирования на компьютере. Построенный в данной программе чертёж получается динамическим, то есть эта программа позволяет «оживить» полученный рисунок, пронаблюдать, как он «на лету» меняется при перемещении базовых точек мышкой. Таким образом, DG предоставляет модель геометрии на плоскости: геометрических объектов и зависимостей между ними.

Школьникам 7-9 классов, испытывающим трудности при изучении геометрии в школе, данные программы помогут разобраться в основных геометрических понятиях, «разглядеть» сложные чертежи.

С использованием этих программ мною описаны правила построения геометрических фигур, таких как прямая, отрезок, окружность, многоугольник и др. Рассмотрены школьные темы: «Угол», «Треугольник, его медиана, высота, биссектриса», «Окружность, вписанная в треугольник», «Окружность, описанная около треугольника», «Геометрические преобразования» (центральная симметрия, симметрия относительно прямой, параллельный перенос, гомотетия, поворот) и др.

Моей задачей является разработка методических пособий и поурочных планов с использованием этих электронных учебников, что послужит толчком к проведению уроков с применением компьютера, а, следовательно, к повышению интереса учащихся к предмету и качества усвоения учебного материала.

## Система $M/M/1$ с отключением сервера

*Бабчинская Е. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

Рассмотрим систему массового обслуживания  $M/M/1$ , которая находится в состоянии статистического равновесия. Сервер работает непрерывно, пока

в системе имеется хотя бы одна заявка. Когда сервер заканчивает обслуживание заявки и обнаруживает систему пустой, он отключается на случайное время  $\gamma$ , распределенное по показательному закону, которое называется периодом отключения. В конце периода отключения сервер включается и начинает обслуживать те заявки, которые поступили во время периода отключения сервера. Если сервер не обнаружит ожидающих заявок в конце периода отключения, он сразу же уходит еще на один период отключения, и продолжает действовать таким образом до тех пор, пока не обнаружит хотя бы одну ожидающую заявку. Требуется найти стационарные характеристики этой системы. Проиллюстрируем данную задачу диаграммой переходов из состояния

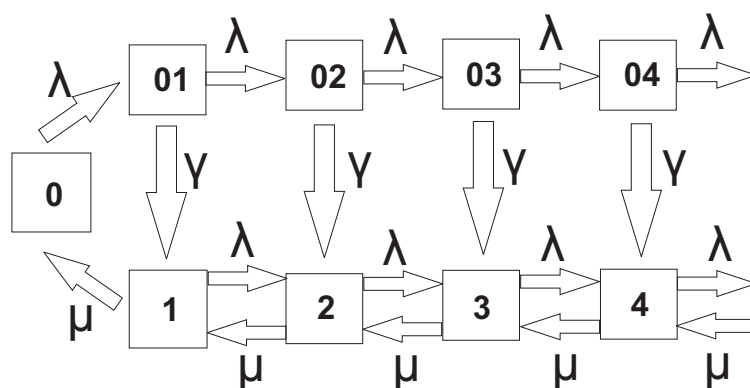


Рис. 1. Диаграмма переходов системы

в состояние. Здесь состояние  $E_{0k} = \{\text{в системе } k \text{ заявок, сервер отключен}\}$ ,  $E_k = \{\text{в системе } k \text{ заявок, сервер работает}\}$ . Пусть  $\lambda$  — интенсивность поступления заявок,  $\mu$  — интенсивность обслуживания заявок,  $\gamma$  — параметр показательного распределения случайного времени отключения. Обозначим через  $\Pi_{0k}$  вероятность того, что система находится в состоянии  $E_{0k}$ ,  $\Pi_k$  — вероятность того, что система находится в состоянии  $E_k$ . И пусть  $g_0(z) = \sum_{k=1}^{\infty} \Pi_{0k} z^k$  и  $g_1(z) = \sum_{k=0}^{\infty} \Pi_k z^k$  — соответствующие производящие функции этих вероятностей. А общая производящая функция системы имеет вид:

$$G(z) = g_0(z) + g_1(z),$$

при этом  $G(1) = 1$  — условие нормировки.

Составим систему уравнений равновесия "по стрелочкам":

$$\left\{ \begin{array}{l} \lambda\Pi_{01} + \gamma\Pi_{01} = \lambda\Pi_0 \\ \lambda\Pi_{02} + \gamma\Pi_{02} = \lambda\Pi_{01} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \lambda\Pi_{0n} + \gamma\Pi_{0n} = \lambda\Pi_{0n-1} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \end{array} \right. \quad (1)$$

$$\left\{ \begin{array}{l} \lambda\Pi_0 = \mu\Pi_1 \\ \lambda\Pi_1 + \mu\Pi_1 = \gamma\Pi_{01} + \mu\Pi_2 \\ \lambda\Pi_2 + \mu\Pi_2 = \gamma\Pi_{02} + \mu\Pi_3 + \lambda\Pi_1 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \lambda\Pi_n + \mu\Pi_n = \gamma\Pi_{0n} + \mu\Pi_{n+1} + \lambda\Pi_{n-1} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \end{array} \right. \quad (2)$$

Умножим каждое уравнение системы (5) на  $z, z^2, z^3, \dots, z^n$  соответственно и просуммируем.

$$\lambda \sum_{n=1}^{\infty} \Pi_{0n} z^n + \gamma \sum_{n=1}^{\infty} \Pi_{0n} z^n = z \lambda \Pi_0 + \lambda \sum_{n=2}^{\infty} \Pi_{0n-1} z^n$$

Так как  $g_0(z) = \sum_{k=1}^{\infty} \Pi_{0k} z^k$ , то

$$\lambda g_0(z) + \gamma g_0(z) = z \lambda \Pi_0 + \lambda z g_0(z),$$

$$g_0(z)(\lambda + \gamma - \lambda z) = \lambda z \Pi_0,$$

$$g_0(z) = \Pi_0 \frac{\lambda z}{\lambda + \gamma - \lambda z}.$$

Умножим каждое уравнение системы (2) на  $z^0, z^1, z^2, z^3, \dots, z^n$  соответственно и просуммируем:

$$\lambda \sum_{n=0}^{\infty} \Pi_n z^n + \mu \sum_{n=1}^{\infty} \Pi_n z^n = \gamma \sum_{n=1}^{\infty} \Pi_{0n} z^n + \mu \sum_{n=0}^{\infty} \Pi_{n+1} z^n + \lambda \sum_{n=2}^{\infty} \Pi_{n-1} z^n.$$

Так как  $g_0(z) = \sum_{k=1}^{\infty} \Pi_{0k} z^k$  и  $g_1(z) = \sum_{k=0}^{\infty} \Pi_k z^k$ , то

$$\lambda g_1(z) + \mu(g_1(z) - \Pi_0) = \gamma g_0(z) + \frac{\mu}{z}(g_1(z) - \Pi_0) + \lambda z(g_1(z) - \Pi_0),$$

Сгруппируем члены равенства и подставим найденное  $g_0(z)$ :

$$g_1(z) \left( \lambda + \mu - \frac{\mu}{z} - \lambda z \right) = \Pi_0 \left( \mu + \gamma \frac{\lambda z}{\lambda + \gamma - \lambda z} - \frac{\mu}{z} - \lambda z \right).$$

$$g_1(z) = \Pi_0 \frac{\mu + \frac{\gamma \lambda z}{\lambda + \gamma - \lambda z} - \frac{\mu}{z} - \lambda z}{\lambda + \mu - \frac{\mu}{z} - \lambda z} = \Pi_0 \frac{\mu z + \frac{\gamma \lambda z^2}{\lambda + \gamma - \lambda z} - \mu - \lambda z^2}{(z-1)(\mu - \lambda z)}.$$



$$g_1(z) = \Pi_0 \frac{\mu + \frac{\lambda^2 z^2}{\lambda + \gamma - \lambda z}}{\mu - \lambda z}.$$

Имеем:

$$G(z) = g_0(z) + g_1(z) = \Pi_0 \frac{\lambda z}{\lambda + \gamma - \lambda z} + \Pi_0 \frac{\mu + \frac{\lambda^2 z^2}{\lambda + \gamma - \lambda z}}{\mu - \lambda z}.$$

Используя условие  $G(1) = 1$ , найдем  $\Pi_0$ :

$$\begin{aligned} \Pi_0 \frac{\lambda}{\lambda + \gamma - \lambda} + \Pi_0 \frac{\mu + \frac{\lambda^2}{\lambda + \gamma - \lambda}}{\mu - \lambda} &= 1 \\ \Pi_0 &= \frac{\gamma(\mu - \lambda)}{\mu(\lambda + \gamma)}. \end{aligned}$$

Подставляя найденное  $\Pi_0$ , найдём окончательно общую производящую функцию:

$$G(z) = \frac{\gamma(\mu - \lambda)}{(\lambda + \gamma - \lambda z)(\mu - \lambda z)}$$

Найдём вероятность того, что линия отключена:

$$g_0(1) + \Pi_0 = \frac{\gamma(\mu - \lambda)}{\mu(\lambda + \gamma)} \left( \frac{\lambda}{\gamma} + 1 \right) = 1 - \frac{\lambda}{\mu}.$$

Вычислим вероятность того, что линия работает:

$$g_1(1) - \Pi_0 = \frac{\gamma(\mu - \lambda)}{\mu(\lambda + \gamma)} \left( \frac{\mu + \frac{\lambda^2}{\gamma}}{\mu - \lambda} - 1 \right) = \frac{\gamma(\mu - \lambda)}{\mu(\lambda + \gamma)} \cdot \frac{\lambda(\lambda + \gamma)}{\gamma(\mu - \lambda)} = \frac{\lambda}{\mu}.$$

Среднее количество заявок в системе будет равно:

$$L = G'(1) = \frac{\gamma(\mu - \lambda)(\lambda(\mu - \lambda) + \lambda\gamma)}{\gamma^2(\mu - \lambda)^2} = \frac{\lambda\mu - \lambda^2 + \lambda\gamma}{\gamma(\mu - \lambda)} = \frac{\lambda}{\gamma} + \frac{\lambda}{\mu - \lambda}.$$

Среднее время пребывания заявки в системе можно найти по формуле Литтла:

$$W = \frac{L}{\lambda} = \frac{1}{\gamma} + \frac{1}{\mu - \lambda}.$$

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Robert B.Cooper *Introduction to Queueing Theory. Second Edition.* – New York: North Holland, – 1981.
- [2] Клейнрок Б.Д. *Теория массового обслуживания.* – М.: Машиностроение, – 1975.
- [3] Бочаров П.П., Печинкин А.В. *Теория массового обслуживания.* – М.: Изд-во РУДН, – 1995 – С. 529.

## Некоторые вопросы бесконечномерного анализа

*Безруков М. Н.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

Рассматриваются дифференциальные операторы, зависящие от конечного набора векторных полей, определенные на гладких функциях равенством

$$D^{m+1}\varphi(h_1, \dots, h_{m+1}) = D_{h_{m+1}}D^m\varphi(h_1, \dots, h_m) - \sum_{i=1}^n D^m(h_1, \dots, D_{h_{m+1}}h_i, \dots, h_m),$$

где  $D^1\varphi(h) = L_h\varphi = \frac{d}{dt} \Big|_{t=0} \varphi(S_h^t x)$  – производная функции  $\varphi$  вдоль векторного поля  $h$  ( $S_h^t x$  – интегральный поток векторного поля  $h$ ),

$L_{h_1}h_2$  – производная векторного поля  $h_2$  вдоль векторного поля  $h_1$ , определяемая аналогично.

Устанавливаются свойства таких операторов, доказывается, что оператор  $D^m$  является полилинейным, при этом свойство полилинейности оказывается верным не только для умножения на число, но и для умножения на гладкую функцию. Кроме этого оператор  $D^m$  симметричен относительно аргументов  $h_1, \dots, h_m$ .

Оператор  $D^m$  естественно назвать производной порядка  $m$  вдоль конечного набора векторных полей  $h_1, \dots, h_m$ .

При помощи этих операторов можно вычислять интегралы по конечномерным и бесконечномерным пространствам, применение указанного метода вычисления интегралов является одним из вариантов обобщения классической формулы интегрирования по частям, известным из курса математического анализа. В частности, для вещественного сепарабельного гильбертова пространства с гауссовой мерой легко получаются хорошо известные формулы.

### СПИСОК ЛИТЕРАТУРЫ

- [1] Ю. Л. Далецкий, Я. И. Белополюская *Стохастические уравнения и дифференциальная геометрия.* // К.: Вища школа, 1989. - 296 с.
- [2] В. И. Арнольд *Математические методы классической механики.* // М.: Наука, 1989. - 472 с.

## Полные линейные интегродифференциальные уравнения второго порядка в гильбертовом пространстве (параболический случай)

*Гажева И.А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

В произвольном комплексном гильбертовом пространстве  $\mathcal{H}$  рассматривается задача Коши

$$\frac{d^2u}{dt^2} + (F + iG)\frac{du}{dt} + Bu + \sum_{k=1}^m \int_0^t G_k(t, s) A_k u(s) ds = f(t), \quad (1)$$

$$u(0) = u^0, \quad u'(0) = u^1, \quad (2)$$

для полного линейного интегродифференциального уравнения второго порядка, где  $u = u(t)$  - искомая функция переменной  $t$  со значениями в гильбертовом пространстве  $\mathcal{H}$ ,  $f(t)$  - заданная функция, а операторы  $B$ ,  $F$  и  $G$  имеют отчетливый физический смысл. К задачам такого типа приводит проблема малых движений идеальной релаксирующей жидкости в произвольной ограниченной области. Искомая функция представляет собой поле смещений жидкости от положения равновесия.

Рассматривается задача, разрешенная относительно старшей производной, потому единичный оператор, стоящий перед второй производной, является оператором кинетической энергии. Оператор  $B$  является оператором потенциальной энергии малых движений системы относительно состояния равновесия, он самосопряжен и в состоянии устойчивого равновесия неотрицателен:  $B = B^* \geq 0$ . Оператор  $F$  играет роль оператора диссипации энергии, поэтому далее предполагается, что  $F = F^* \geq 0$ . Оператор  $G = G^*$  - гироскопический (кориолисов) оператор, который считается ограниченным:  $G \in \mathcal{L}(\mathcal{H})$ .

Оператор-функции  $G_k(t, s)$  предполагаются ограниченными, действующими в гильбертовом пространстве  $\mathcal{H}$ ,  $A_k$  - операторы, действующие в  $\mathcal{H}$ ,  $k = \overline{1, m}$ .

Параболический случай предполагает доминирование оператора диссипации  $F$ , что соответствует ситуации, когда диссипативные силы достаточно велики:

$$\mathcal{D}(F) \subset \mathcal{D}(B). \quad (3)$$

**Определение 1.** Функция  $u = u(t)$  со значениями в  $\mathcal{H}$  называется сильным решением задачи Коши (1), (2) на промежутке  $[0, T]$ , если выполнены следующие условия:

- а)  $u(t) \in \mathcal{D}(B)$  для любого  $t \in [0, T]$  и  $Bu(t) \in C([0, T]; \mathcal{H})$ ;
- б)  $\frac{du}{dt} \in \mathcal{D}(F)$  и  $F \frac{du}{dt} \in C([0, T]; \mathcal{H})$ ;
- в)  $u(t) \in C^2([0, T]; \mathcal{H})$ ;
- г) выполнено уравнение (1) при  $t \in [0, T]$ ;
- д) выполнены начальные условия (2).

Переносим интегральный член вправо, вводя

$$\hat{f}(t) = f(t) - \sum_{k=1}^m \int_0^t G_k(t, s) A_k u(s) ds \quad (4)$$

и осуществляя серию замен

$$\frac{dv}{dt} = -iB^{1/2}u(t), \quad v(0) = 0, \quad (5)$$

$$y(t) = (u'(t); v'(t))^T, \quad (6)$$

$$\mathcal{A}_0 := \begin{pmatrix} F + iG & iB^{1/2} \\ iB^{1/2} & 0 \end{pmatrix}, \quad \mathcal{D}(\mathcal{A}_0) := \mathcal{D}(F) \oplus \mathcal{D}(B^{1/2}). \quad (7)$$

$$y(t) = e^{at}z(t), \quad a > 0, \quad (8)$$

приходим к задаче

$$\frac{dz}{dt} + \mathcal{A}_a z = f_a(t), \quad z(0) = y^0, \quad (9)$$

$$\mathcal{A}_a = \begin{pmatrix} F_a + iG & iB^{1/2} \\ iB^{1/2} & aI \end{pmatrix}, \quad F_a := F + aI, \quad f_a(t) = e^{-at} (\hat{f}(t); 0)^T, \quad (10)$$

где операторная матрица  $\mathcal{A}_a$  равномерно аккретивна.

Введение вспомогательных операторов

$$Q_a := B^{1/2}F_a^{-1/2}, \quad Q_a^+ := F_a^{-1/2}B^{1/2}, \quad \mathcal{D}(Q_a^+) := \mathcal{D}(B^{1/2}). \quad (11)$$

позволяет получить факторизацию операторной матрицы  $\mathcal{A}_a$  и осуществить ее замыкание.

**Теорема 1.** Операторная матрица  $\mathcal{A}_a$ , заданная на области определения  $\mathcal{D}(\mathcal{A}_a) = \mathcal{D}(F) \oplus \mathcal{D}(B^{1/2})$ , имеет следующие представления:

1<sup>0</sup>. в форме Шура - Фробениуса

$$\mathcal{A}_a = \begin{pmatrix} I & 0 \\ iQ_a F_a^{-1/2} & I \end{pmatrix} \begin{pmatrix} F_a & 0 \\ 0 & aI + Q_a Q_a^+ \end{pmatrix} \begin{pmatrix} I & iF_a^{-1/2} Q_a^+ \\ 0 & I \end{pmatrix} + \begin{pmatrix} iG & 0 \\ 0 & 0 \end{pmatrix}; \quad (12)$$

2<sup>0</sup>. с симметрическими крайними множителями

$$\mathcal{A}_a = \begin{pmatrix} F_a^{1/2} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I + iF_a^{-1/2} G F_a^{-1/2} & iQ_a^+ \\ iQ_a & aI \end{pmatrix} \begin{pmatrix} F_a^{1/2} & 0 \\ 0 & I \end{pmatrix}. \quad (13)$$

Оператор  $\mathcal{A}_a$  замыкаем и его замыкание  $\mathcal{A} := \overline{\mathcal{A}_a}$  является максимальным равномерно аккретивным оператором:

$$\operatorname{Re}(\mathcal{A}z, z)_{\mathcal{H}^2} \geq a \|z\|_{\mathcal{H}^2}^2, \quad z \in \mathcal{D}(\mathcal{A}). \quad (14)$$

Он допускает два представления:

1<sup>0</sup>. в форме Шура - Фробениуса

$$\mathcal{A} = \begin{pmatrix} I & 0 \\ iQ_a F_a^{-1/2} & I \end{pmatrix} \begin{pmatrix} F_a & 0 \\ 0 & aI + Q_a Q_a^* \end{pmatrix} \begin{pmatrix} I & iF_a^{-1/2} Q_a^* \\ 0 & I \end{pmatrix} + \begin{pmatrix} iG & 0 \\ 0 & 0 \end{pmatrix}; \quad (15)$$

2<sup>0</sup>. с симметрическими крайними множителями

$$\mathcal{A}_a = \begin{pmatrix} F_a^{1/2} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I + iF_a^{-1/2} G F_a^{-1/2} & iQ_a^* \\ iQ_a & aI \end{pmatrix} \begin{pmatrix} F_a^{1/2} & 0 \\ 0 & I \end{pmatrix}. \quad (16)$$

Оператор  $\mathcal{A}$  задан на области определения

$$\mathcal{D}(\mathcal{A}) := \{z = (z_1; z_2)^T \in \mathcal{H}^2 : z_1 + iF_a^{-1/2} Q_a^* z_2 \in \mathcal{D}(F_a)\} \quad (17)$$

и действует на ней по закону

$$\mathcal{A}z = \begin{pmatrix} F_a (z_1 + iF_a^{-1/2} Q_a^* z_2) + iG z_1 \\ iQ_a F_a^{1/2} z_1 + a z_2 \end{pmatrix} = \begin{pmatrix} F_a^{1/2} (F_a^{1/2} z_1 + iQ_a^* z_2) + iG z_1 \\ iQ_a F_a^{1/2} z_1 + a z_2 \end{pmatrix}. \quad (18)$$

**Следствие 1.** Операторная матрица  $(-\mathcal{A})$  является генератором сжимающей полугруппы  $U(t)$ , причем

$$\|U(t)\| \leq e^{-at}, \quad t \geq 0. \quad (19)$$

Далее используется теорема о сильной разрешимости задачи Коши для интегродифференциального уравнения первого порядка

$$\frac{dz}{dt} + \mathcal{A}z = f_a(t), \quad z(0) = y^0 = (u^1; -iB^{1/2}u^0)^T \quad (20)$$

и с помощью теории полугрупп осуществляется переход к интегральному уравнению Вольтерра второго рода, которое рассматривается в гильбертовом пространстве с нормой графика. При условиях

$$\mathcal{D}(A_k) \supset \mathcal{D}(F^{1/2}), \quad G_k(t, s) \in C(\Delta_T, \mathcal{L}(\mathcal{H})), \quad \frac{\partial G_k}{\partial t} \in C(\Delta_T, \mathcal{L}(\mathcal{H})), \quad (21)$$

$$k = \overline{1, m}, \quad \Delta_T = \{(t, s) : 0 \leq s \leq t \leq T\}$$

существует сильное решение этого уравнения. Осуществление обратного хода позволяет получить итоговый результат:

**Теорема 2.** Пусть в задаче (1) - (3) выполнены условия (21),

$$u^0 \in \mathcal{D}(B), \quad u^1 \in \mathcal{D}(F), \quad (22)$$

а также одно из условий:

$$f(t) \in W_p^1([0, T]; \mathcal{H}), \quad p > 1, \quad (23)$$

$$F^{-1} \in \mathfrak{S}_\infty, \quad f(t) \in C^\alpha([0, T]; \mathcal{H}), \quad 0 < \alpha \leq 1. \quad (24)$$

Тогда следующие условия эквивалентны:

- 1<sup>0</sup>. Задача (1) - (3) имеет единственное сильное решение на отрезке  $[0, T]$ .
- 2<sup>0</sup>. Задача (9) - (10) имеет единственное сильное решение на отрезке  $[0, T]$ .
- 3<sup>0</sup>. Задача Коши (20) имеет единственное сильное решение на отрезке  $[0, T]$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Копачевский Н. Д. Интегродифференциальные уравнения Вольтерра в банаховом пространстве. 2004 г. (рукопись)
- [2] Копачевский Н. Д. Операторные методы математической физики: Специальный курс лекций. – Симферополь: ООО «ФОРМА», 2008. – 140 с.
- [3] N.D.Kopachevskii, S.G.Krein. Operator Approach in Linear Problems of Hydrodynamics. Vol. 1: Self-Adjoint Problems for Ideal Fluid; Vol. 2: Not Self-Adjoint Problems for Viscous Fluid. - Birkhauser Vertlad, Basel, Boston, Berlin, 1999.

## Горение вдоль полосы

Кандагура А. Н.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ДИФФЕРЕНЦИАЛЬНЫХ И ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ (ГРУППА 500)

Рассмотрим уравнение:

$$\ddot{\xi} + \xi = 2\varepsilon[\dot{\xi}(1 - \frac{4}{3}\dot{\xi}^2) + \frac{\lambda^2}{4\pi^2}\Delta\dot{\xi} + \frac{\beta\lambda}{2\pi}\sqrt{-\Delta\dot{\xi}}], \quad (1)$$

с условиями Дирихле:

$$\dot{\xi}|_{x=0} = \dot{\xi}|_{x=l} = 0, \quad (2)$$

где  $\lambda > 0$  – корреляционная длина теплопроводностной связи соседних участков фронта,  $\beta > 0$  – коэффициент нелокальной связи участков фронта, а  $0 < \varepsilon \ll 1$  – инкремент неустойчивости.

Величина  $\xi$  представляет собой отклонение фронта горения от невозмущенного, соответствующего стационарному режиму. Она зависит от времени  $t$  и от координаты  $x$ . Точка означает дифференцирование по времени,  $\Delta$  – одномерный лапласиан. Данная задача описывает безгазовое горение вдоль теплоизолированной полосы шириной  $l$ .

Данная задача (1)-(2) была исследована в [2] при  $\beta = 0$ . С помощью метода Галеркина было найдено простейшее приближенное решение в виде стоячей волны:

$$\xi = \sqrt{\frac{4}{3}\left(1 - \frac{\lambda^2}{4l^2}\right)} \sin t \sin \frac{\pi x}{l}.$$

Как отмечается в [2], указанное решение устойчиво при  $l < \lambda$ .

Исследование уравнения (1) проводилось [1] в случае горения на цилиндре:

$$\xi(t, x + 2\pi k) = \xi(t, x).$$

Изучению режимов, возникающих при рассмотрении уравнения (1) с различными условиями также посвящены работы [3], [5].

Задачу (1)-(2) будем рассматривать в фазовом пространстве  $E = \{(\xi, \dot{\xi}), \xi, \dot{\xi} \in \mathbf{H}^1(o; l)\}$ . Указанная задача на пространстве  $E$  порождает непрерывную полугруппу.

Для отыскания приближенного решения задачи (1)-(2) воспользуемся методом Галёркина. Обобщая указанный метод, решение будем искать в виде:

$$\xi = \sum_{k=1}^n (z_k + \bar{z}_k) \sin \frac{\pi k}{l} x + \varepsilon p_3(z, \bar{z}, x) \quad (3)$$

где  $z_k$ ,  $k = 1..n$  удовлетворяют системе

$$\dot{z}_k = (i + \varepsilon\alpha_k)z_k + \varepsilon f_k(z, \bar{z}). \quad (4)$$

Переменная  $\bar{z}$  удовлетворяет соответствующему комплексно сопряженному уравнению.

Здесь  $p_3, f_k, k = 1..n$  - формы третьей степени относительно  $z$ ,

$$\alpha_k = 1 - k^2\mu^2 + \beta k\mu,$$

где  $\mu = \frac{\lambda}{2l}$ .

Подставив (3) с учётом (4), получим:

$$\frac{1}{i}\mathbf{L}p_3 + F = \frac{8}{3} \left( \sum_{k=1}^n (z_k - \bar{z}_k) \sin \frac{\pi k x}{l} \right)^3, \quad (5)$$

где

$$\mathbf{L}p_3 = p_3 + \sum_{s+r=2} (-1)^r \frac{\partial^2 p_3}{\partial z_j^s \partial \bar{z}_k^r} z_j^s \bar{z}_k^r, \quad s = 0, 1, 2, r = 0, 1, 2, j = 1..n, k = 1..n,$$

$$F = \sum_{k=1}^n \left( f_k - \bar{f}_k + \sum_{j=1}^n \left( \frac{\partial f_k}{\partial z_j} z_j - \frac{\partial f_k}{\partial \bar{z}_j} \bar{z}_j + \frac{\partial \bar{f}_k}{\partial z_j} z_j - \frac{\partial \bar{f}_k}{\partial \bar{z}_j} \bar{z}_j \right) \right) \sin \frac{\pi k}{l} x.$$

Оператор  $\mathbf{L}$  является диагональным на пространстве многочленов относительно  $z, \bar{z}$ . При этом:

$$\mathbf{L}z^\alpha \bar{z}^\beta = (1 - |\alpha - \beta|^2) z^\alpha \bar{z}^\beta.$$

Система (5) даёт возможность составить систему обыкновенных дифференциальных уравнений относительно  $z_k$ . Исследуя данную систему на наличие стационарных точек и устойчивость по первому приближению, можно судить о характере и качественном поведении решений исходной задачи (1)-(2).

Таким образом, вместо исходной задачи (1)–(2) будем исследовать систему из  $2n$  дифференциальных уравнений.

Рассмотрим случаи двух- и четырёхмодовой аппроксимации подробнее.

В случае  $n = 1$  (двухмодовая аппроксимация) система дифференциальных уравнений будет иметь вид:

$$\begin{aligned} \dot{z}_1 &= iz_1 + \varepsilon(\alpha_1 z_1 - \frac{3}{4} z_1^2 \bar{z}_1) \\ \dot{\bar{z}}_1 &= i\bar{z}_1 + \varepsilon(\alpha_1 \bar{z}_1 - \frac{3}{4} z_1^2 \bar{z}_1) \end{aligned} \quad (6)$$

Осуществим преобразование  $z_1 \rightarrow a_1 e^{it}, t = \frac{\tau}{\varepsilon}$ . Тогда система примет вид:

$$\begin{aligned} \dot{a}_1 &= a_1(\alpha_1 - \frac{3}{4}|a_1|^2), \\ \dot{\bar{a}}_1 &= \bar{a}_1(\alpha_1 - \frac{3}{4}|a_1|^2) \end{aligned} \quad (7)$$



Исследуя систему (7) на наличие стационарных точек и их устойчивость по первому приближению, выясняем, что данная система имеет стационарную точку  $a_1 = 0$  и при  $\alpha_1 > 0$  целую окружность стационарных точек  $|a_1| = 2\sqrt{\frac{\alpha_1}{3}}$ . Точке  $a_1 = 0$  и соответствует решение исходной задачи (1)–(2) вида  $\xi = O(\varepsilon)$ . Это решение устойчиво только тогда, когда параметр  $\alpha_1 < 0$ . Когда  $\alpha_1 > 0$  существует и устойчиво решение  $\xi = 2\sqrt{\frac{\alpha_1}{3}} \sin t \sin \frac{\pi x}{l} + O(\varepsilon)$ .

Будем интересоваться поведением системы дифференциальных уравнений в случае 4-модовой аппроксимации задачи (1)–(2).

В этом случае система примет вид:

$$\begin{aligned} \dot{z}_1 &= iz_1 + \varepsilon z_1(\alpha_1 - 3|z_1|^2 - 4|z_2|^2) - 2\bar{z}_1 z_2^2, \\ \dot{z}_2 &= iz_2 + \varepsilon z_2(\alpha_2 - 4|z_1|^2 - 3|z_2|^2) - 2z_1^2 \bar{z}_2. \end{aligned} \quad (8)$$

В данной системе выполним преобразование:  $z_k \rightarrow a_k e^{it}$ ,  $k = 0, 1$  и при этом осуществим переход к медленной переменной  $t = \frac{\tau}{\varepsilon}$ . Получим:

$$\begin{aligned} \dot{a}_1 &= a_1(\alpha_1 - \frac{3}{4}|a_1|^2 - |a_2|^2) - \frac{1}{2}\bar{a}_1 a_2^2, \\ \dot{a}_2 &= a_2(\alpha_2 - |a_1|^2 - \frac{3}{4}|a_2|^2) - \frac{1}{2}a_1^2 \bar{a}_2. \end{aligned} \quad (9)$$

Если параметр  $\mu$  находится в области  $\mu < \frac{\beta + \sqrt{\beta^2 + 4}}{4}$ , система (9) будет иметь две окружности стационарных точек  $a_1 = 0, |a_2| = 2\sqrt{\frac{\alpha_2}{3}}$ . Данным точкам соответствует решение задачи (1)–(2) вида  $\xi = 2\sqrt{\frac{\alpha_2}{3}} \sin t \sin \frac{2\pi x}{l} + O(\varepsilon)$ . С помощью теоремы Ляпунова об устойчивости по первому приближению было выяснено, что указанные точки и соответствующие решения устойчивы, когда параметр  $\mu$  находится в пределах:  $\mu \in \left( \frac{\beta - \sqrt{\beta^2 - 20}}{10}; \frac{\beta - \sqrt{\beta^2 - 20}}{10} \right)$ .

Когда бифуркационный параметр  $\mu$  переходит значение  $\mu = \frac{\beta + \sqrt{\beta^2 + 4}}{4}$ , рождаются ещё две окружности стационарных точек  $|a_1| = \sqrt{\frac{\alpha_1}{3}}, a_2 = 0$ . Этим точкам отвечают решения  $\xi = 2\sqrt{\frac{\alpha_1}{3}} \sin t \sin \frac{\pi x}{l} + O(\varepsilon)$ . Указанные окружности стационарных точек и решения устойчивы в области значения параметра  $\mu < \frac{2\beta + \sqrt{4\beta^2 + 10}}{10}$ .

Однако следует отметить, что в случае, когда на параметр  $\beta$  будет наложено специальное условие, а именно:  $\beta^2 > 20$ , возможно существование следующих окружностей стационарных точек и соответствующих им решений. Это  $|a_1| = 2\sqrt{\frac{3\alpha_1 - 2\alpha_2}{5}}, |a_2| = 2\sqrt{\frac{3\alpha_2 - 2\alpha_1}{5}}$ . Данным окружностям особых точек отвечают такие решения исходной задачи (1)–(2):  $\xi = 2\sqrt{\frac{3\alpha_1 - 2\alpha_2}{5}} \sin t \sin \frac{\pi x}{l} + 2\sqrt{\frac{3\alpha_2 - 2\alpha_1}{5}} \sin t \sin \frac{2\pi x}{l} + O(\varepsilon)$ . Указанное возможно только в том случае, когда  $\mu$  принимает значения

из области:  $\mu \in [0; \frac{\beta - \sqrt{\beta^2 - 20}}{10}] \cup [\frac{\beta + \sqrt{\beta^2 - 20}}{10}; \frac{2\beta + \sqrt{4\beta^2 + 10}}{10}]$ . И при значениях  $\mu < \frac{3\beta + \sqrt{9\beta^2 + 28}}{14}$  существуют окружности стационарных точек вида:  $|a_1| = \frac{2}{3}\sqrt{2\alpha_2 - \alpha_1}$ ,  $|a_2| = \frac{2}{3}\sqrt{2\alpha_1 - \alpha_2}$ . Им отвечают решения исходной задачи (1)-(2)  $\xi = \frac{2}{3}\sqrt{2\alpha_2 - \alpha_1} \sin t \sin \frac{\pi x}{l} + \frac{2}{3}\sqrt{2\alpha_1 - \alpha_2} \sin t \sin \frac{2\pi x}{l}$ .

Исследование поведения решений задачи (1)-(2) в случае более чем 4-модовой аппроксимации возможно с помощью лишь численных расчётов.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Самойленко А.М., Белан Е.П. *Динамика бегущих волн феноменологического уравнения спинового горения.* // Доклады Академии наук. – Т. 406, № 6. –2006. – С. 138-141.
- [2] Зельдович Я.Б., Маломед Б.А. *название.* //Изв. вузов. Радиофизика. – Т. 15, № 6. – 1982. – С. 591-618.
- [3] Маломед Б.А. *Распространение автоколебательных волн вдоль полосы.* // Изв. вузов. Радиофизика. – Т. 14, № 5. – 1981. – С. 571-576.
- [4] Агладзе К.И., Давыдов В.А., Михайлов А.С. *название.* // Письма в ЖЭТФ. – Т. 45, № 12. – С. 601-603.
- [5] Алдушин А.П., Маломед Б.А. *Феноменологическое описание нестационарных неоднородных волн горения.* // Физика горения и взрыва. – Т. 17, №1. –1981. – С. 3-12.

## Вольтеррово интегродифференциальное уравнение второго порядка (гиперболический случай)

*Лапицкая Д. С.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

В гильбертовом пространстве  $H$  рассмотрим задачу Коши для вольтеррова интегродифференциального уравнения второго порядка следующего вида

$$A \frac{d^2 u}{dt^2} + (F + iK) \frac{du}{dt} + Bu + \sum_{k=1}^m \int_0^t G_k(t, s) A_k u(s) ds = f(t),$$

$$u(0) = u^0, \quad u'(0) = u^1.$$

Здесь  $u = u(t)$  - искомая функция,  $f = f(t)$  - заданная функция,  $A_0, A_1, \dots, A_m$  - операторы, вообще говоря, неограниченные и действующие в  $H$ , а  $G_k(t, s)$  - оператор-функции  $t, s \in \mathbb{R}$  со значениями во множестве  $\mathcal{L}(H)$  линейных ограниченных операторов, действующих в  $H$ .

Операторы  $A$ ,  $F$ ,  $K$  и  $B$  имеют отчетливый физический смысл.

$A$  - оператор кинетической энергии, т.к. квадратичная форма  $\frac{1}{2} (A \frac{du}{dt}, \frac{du}{dt})$  есть кинетическая энергия системы, поэтому естественно предполагать  $A = A^* > 0$ . Предполагается, что  $A$  - обратимый, хотя обратный может быть и неограниченным.

$B$  - оператор потенциальной энергии. Далее, квадратичная форма  $\frac{1}{2} (B \frac{du}{dt}, \frac{du}{dt})$  - потенциальная энергия малых движений системы относительно состояния равновесия, поэтому далее предполагается, что оператор потенциальной энергии  $B$  - самосопряжен и в состоянии устойчивого равновесия неотрицателен:

$$B = B^* \geq 0.$$

$F$ - оператор диссипации энергии,  $(F \frac{du}{dt}, \frac{du}{dt})$  - скорость диссипации энергии. Поэтому будем считать, что  $F = F^* \geq \gamma_F I$ ,  $\gamma_F \in \mathbb{R}$ .

$K = K^*$  - гидроскопический оператор, который обычно является ограниченным (но может быть неограниченным), он возникает при движении системы в неинерциальных системах координат.

Рассмотрим частный случай:

$$A = I, \quad K = 0, \quad F \geq 0, \quad B \gg 0,$$

тогда исходная задача Коши примет вид:

$$\begin{aligned} \frac{d^2 u}{dt^2} + F \frac{du}{dt} + Bu + \sum_{k=1}^m \int_0^t G_k(t, s) A_k u(s) ds &= f(t), \\ u(0) = u^0, \quad u'(0) = u^1, \end{aligned} \quad (1)$$

а также пусть выполнено условие сравнения:

$$\mathcal{D}(B^{\frac{1}{2}}) \subset \mathcal{D}(F). \quad (2)$$

Из (2) следует условие  $\mathcal{D}(B) \subset \mathcal{D}(F)$ , характерное для гиперболического случая, где доминирующим операторным коэффициентом является  $B$ .

Преобразуем уравнение (1), перенесем сумму в правую часть и выражение, полученное в правой части, переобозначим через  $f_1(t)$ , получим

$$\frac{d^2 u}{dt^2} + F \frac{du}{dt} + Bu = f_1(t), \quad (3)$$

Пусть задача (1)-(3) имеет сильное решение.

Необходимым условием существования сильного решения (1)-(3) на отрезке  $[0, T]$  является условия

$$u^0 \in \mathcal{D}(B), u^1 \in \mathcal{D}(F), f_1(t) \in C([0, T]; H).$$

Одним из основных методов изучения задачи Коши для дифференциального уравнения второго порядка является переход к системе дифференциальных уравнений первого порядка и использование теории сжимающих либо аналитических полугрупп.

Введем новую искомую функцию по закону:

$$\frac{dv}{dt} = -iB^{\frac{1}{2}}u(t), v(0) = 0,$$

и в определении сильного решения потребуем дополнительно, чтобы оно обладало свойством

$$B^{\frac{1}{2}} \frac{du}{dt} \in C([0, T]; H).$$

С учетом этих связей задача (1)-(3) приводится к уравнению

$$\frac{dy}{dt} + \mathcal{A}_0 y = f_2(t), y(0) = y^0, \quad (4)$$

$$y(t) := \left( \frac{du}{dt}; \frac{dv}{dt} \right)^t, y^0 := \left( u^1; -iB^{\frac{1}{2}}u^0 \right)^t,$$

$$\widetilde{A}_k := A_k P_1, P_1 : P_1 y := y_1 = \frac{du}{dt},$$

$$f_2 := \left( f(t) - \sum_{k=1}^m \left( \int_0^t V_k(t, s) \left( \widetilde{A}_k y \right) (s) ds + \int_0^t G_k(t, s) A_k u(0) ds \right); 0 \right)^t,$$

$$\mathcal{A}_0 = \begin{pmatrix} F & iB^{\frac{1}{2}} \\ iB^{\frac{1}{2}} & 0 \end{pmatrix}, \mathcal{D}(\mathcal{A}_0) = \mathcal{D}(B^{\frac{1}{2}}) \oplus \mathcal{D}(B^{\frac{1}{2}}).$$

**Лемма 1.**  $A_0$  - максимальный аккретивный оператор.

**Следствие 1.** Оператор  $(-A_0)$  - генератор  $C_0$  - полугруппы сжимающих операторов.  $U(t)$  - сильно непрерывная полугруппа, отвечающая генератору  $(-A_0)$ .

Можно показать, что задачи (4) и (1) – (3) эквивалентны, тогда из существования и единственности сильного решения (4) следует, что для (1) – (3) также существует единственное сильное решение.

**Теорема 1.** Пусть в задаче Коши (1)-(3) выполнены условия

$$u^0 \in \mathcal{D}(B), \quad u^1 \in \mathcal{D}(B^{\frac{1}{2}}), \quad f(t) \in C^\alpha([0, T]; H), \quad 0 < \alpha \leq 1$$

$$\mathcal{D}(A_k) \supset \mathcal{D}(B^{\frac{1}{2}}), \quad k = (\overline{1, m})$$

$$G_k(t, s), \quad \int_{\xi}^t U(t-s) \frac{\partial G_k}{\partial s}(s, \xi) ds, \quad k = (\overline{1, m})$$

Тогда она имеет единственное сильное решение  $u(t)$  на  $[0, T]$ , причем для этого решения выполнено свойство

$$B^{\frac{1}{2}} \frac{du}{dt} \in C([0, T]; H).$$

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Копачевский Н.Д. *Интегриродифференциальные уравнения Вольтерра в банаховом пространстве.*// Рукопись
- [2] Копачевский Н.Д. *Операторные методы математической физики.* – Специальный курс лекций для студентов специальностей «Математика» и «Прикладная математика». ТНУ, Симферополь, 2008.

## Исследование системы массового обслуживания с ненадежной линией

*Ли Е. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

*В данной статье рассматривается система массового обслуживания с ненадежной линией, строится математическое моделирование, которое в свою очередь приводит к получению вероятностных характеристик системы в стационарном режиме.*

### 1. ПОСТАНОВКА ЗАДАЧИ

Рассматривается система массового обслуживания, состоящая из одной линии и одного места ожидания. На систему поступает простейший поток заявок с интенсивностью  $\lambda$ . Время обслуживания заявки линией является непрерывная случайная величина  $w_1$  с интенсивностью  $\mu_1(x)$ . Время ожидания  $w_2$  обслуживания заявки, находящейся на месте ожидания, экспоненциально распределено с интенсивностью  $\mu_2$ .

Во время обслуживания линия может выходить из строя. Поток отказов линии простейший, с интенсивностью  $\alpha$ . Один наладчик осуществляет ремонт отказавшей линии. Время ремонта непрерывная случайная величина  $w_3$  с интенсивностью  $\mu_3(x)$ .

В правилах обслуживания предполагается, что время затраченное на обслуживание до поломки линии, сохраняется и продолжается после починки ее.

Математическое моделирование данной системы позволяет получить стационарные вероятности для состояний системы.

## 2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Рассмотрим случайный процесс  $\xi(t)$ , фазовое пространство которого состоит из состояний системы  $\{0; 4\}$ .

1. (0)–система исправна.

2. (1,  $w_1$ )–система функционирует, линия обслуживает заявку.  $w_1$ – время обслуживания заявки линией.

Обозначим соответствующую функцию распределения

$$Q_1(t, x) := P\{\xi(t) = 1, w_1 < x\},$$

и плотность распределения

$$q_1(t, x) := \frac{\partial Q_1(t, x)}{\partial x}$$

при этом

$$p_1(t) := Q_1(t, \infty) = \int_0^\infty q_1(t, x) dx = P\{\xi(t) = 1\}$$
–

вероятность нахождения системы в состоянии (1) в момент времени  $t$ .

3. (2,  $w_1, w_2$ )– система функционирует, линия обслуживает заявку и еще одна заявка находится на месте ожидания.  $w_2$ – время ожидания обслуживания заявки.

Обозначим соответствующую функцию распределения

$$Q_2(t, x) := P\{\xi(t) = 2, w_1 < x\},$$

и плотность распределения

$$q_2(t, x) := \frac{\partial Q_2(t, x)}{\partial x}$$

при этом

$$p_2(t) := Q_2(t, \infty) = \int_0^\infty q_2(t, x) dx = P\{\xi(t) = 2\}$$
–

вероятность нахождения системы в состоянии (2) в момент времени  $t$ .

4.  $(3, w_1, w_3)$ – система неисправна, линия ремонтируется, в очереди заявка отсутствует.  $w_3$ – время ремонта.

Обозначим соответствующую функцию и плотность распределения

$$Q_3(t, x, y) := P\{\xi(t) = 3, w_1 < x, w_3 < y\},$$

$$q_3(t, x, y) := \frac{\partial^2 Q_3(t, x, y)}{\partial x \partial y}$$

при этом

$$p_3(t) := Q_3(t, \infty, \infty) = \int_0^\infty dx \int_0^\infty q_3(t, x, y) dy -$$

вероятность нахождения системы в состоянии (3) в момент времени  $t$ .

5.  $(4, w_1, w_2, w_3)$ – система неисправна, линия ремонтируется и в очереди есть заявка.

$$Q_4(t, x, y) := P\{\xi(t) = 4, w_1 < x, w_3 < y\},$$

$$q_4(t, x, y) := \frac{\partial^2 Q_4(t, x, y)}{\partial x \partial y}$$

$$p_4(t) := Q_4(t, \infty, \infty) = \int_0^\infty dx \int_0^\infty q_4(t, x, y) dy -$$

вероятность нахождения системы в состоянии (4) в момент времени  $t$ .

В результате вероятностных рассуждений получили систему интегро-дифференциальных уравнений и граничных условий. Используем свойство эргодичности и перейдем к пределу при  $t \rightarrow \infty$ . Введем обозначения:

$$p_k := \lim_{t \rightarrow \infty} p_k(t), \quad k = \overline{0, 4}$$

$$g_k(\dots) = \lim_{t \rightarrow \infty} q_k(t, \dots), \quad k = \overline{1, 4}$$

В результате получим систему интегро-дифференциальных уравнений и граничных условий для стационарного режима:

$$\lambda p_0 = \int_0^\infty g_1(x) \mu_1(x) dx \quad (1)$$

$$g_1'(x) + (\alpha + \lambda + \mu_1(x))g_1(x) = \mu_2 g_2(x) + \int_0^\infty g_3(x, y) \mu_3(y) dy \quad (2)$$

$$g_2'(x) + (\alpha + \mu_1(x) + \mu_2)g_2(x) = \lambda g_1(x) + \int_0^\infty g_4(x, y)\mu_3(y)dy \quad (3)$$

$$\frac{\partial g_3(x, y)}{\partial y} + (\lambda + \mu_3(y))g_3(x, y) = \mu_2 g_4(x, y) \quad (4)$$

$$\frac{\partial g_4(x, y)}{\partial y} + (\mu_2 + \mu_3(y))g_4(x, y) = \lambda g_3(x, y) \quad (5)$$

$$g_1(0) = \lambda p_0 + \int_0^\infty g_2(x)\mu_1(x)dx \quad (6)$$

$$g_2(0) = 0 \quad (7)$$

$$g_3(x, 0) = \alpha g_1(x) \quad (8)$$

$$g_4(x, 0) = \alpha g_2(x) \quad (9)$$

Заметим, что к системе следует добавить нормировочное равенство

$$\sum_{k=0}^4 p_k = 1.$$

Решая данные уравнения с граничными условиями получили вероятностные характеристики системы  $P_0, P_1, P_2, P_3, P_4$ , а также вероятность потери заявок.

В данной статье была исследована система массового обслуживания с ненадежной линией, выведена система интегро-дифференциальных уравнений с граничными условиями и получены вероятности системы в стационарном режиме.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Коваленко А.И, Марянин Б.Д., Смолич В.П. *Исследование надежности трехэлементной системы с приоритетным обслуживанием двумя наладчиками*
- [2] Анисимов В. В., Закусило О. К., Донченко В. С. *Элементы теории массового обслуживания и асимптотического анализа систем*. Учебное пособие для вузов / Под общей редакцией В. В. Анисимова—1987
- [3] Гнеденко Б. В., Коваленко И. Н. *Введение в теорию массового обслуживания*.

## Статистические пакеты

*Литвиненко А. Л.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
 ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
 КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)



В современных исследованиях к статистическим методам проявляется повышенный интерес как к одному из важнейших аналитических инструментов в сфере поддержки процессов принятия решений.

Большим шагом вперед к развитию статистической науки послужило применение экономико-математических методов и использование компьютерной техники в анализе социально-экономических явлений.

По мере того как из года в год расширяется область приложения математической статистики, заметно возрастает спрос на различного рода руководства по этой дисциплине. Большинство из руководств ориентируется в первую очередь на потребности имеющихся тех или иных специалистов-прикладников (экономистов, агрономов, биологов, медиков, физиков, инженеров-производственников и т. д.).

В данной работе рассмотрено решение одной из распространенных задач математической статистики и приведено её решение с помощью прикладного программного обеспечения MS Office Excel.

**Задача.** В таблице представлен объем спроса на некоторое изделие за последние 10 месяцев.

|   |   |   |   |    |    |    |    |    |    |    |
|---|---|---|---|----|----|----|----|----|----|----|
| X | 1 | 2 | 3 | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| Y | 3 | 5 | 7 | 11 | 13 | 12 | 19 | 23 | 24 | 30 |

Считать, что зависимость между  $x$  и  $y$  имеет вид:

$$y_x = ax + b$$

Написать уравнение регрессии и сделать прогноз объема спроса на следующий месяц  $x = 11$ .

Мы помним, что коэффициент корреляции  $r_{xy}$  находится по формуле:

$$r_{xy} = a \frac{\sigma_y}{\sigma_x} [1],$$

где  $\sigma_y$  и  $\sigma_x$  – выборочные средние квадратичные отклонения.

Отсюда следует, что

$$a = r_{xy} \frac{\sigma_y}{\sigma_x}; \quad (1)$$

$$b = \bar{y} - a\bar{x}. \quad (2)$$

Здесь  $\bar{x}$  и  $\bar{y}$  – выборочное среднее по  $x$  и  $y$  соответственно.

Проведем вспомогательные вычисления в таблице:

| X  | Y  | X <sup>2</sup> | Y <sup>2</sup> |
|----|----|----------------|----------------|
| 1  | 3  | 1              | 9              |
| 2  | 5  | 4              | 25             |
| 3  | 7  | 9              | 49             |
| 4  | 11 | 16             | 121            |
| 5  | 13 | 25             | 169            |
| 6  | 12 | 36             | 144            |
| 7  | 19 | 49             | 361            |
| 8  | 23 | 64             | 529            |
| 9  | 24 | 81             | 576            |
| 10 | 30 | 100            | 900            |

Вычислим  $r_{xy}$ , используя стандартную функцию MS Office Excel **КОРРЕЛ(массивX;массивY)** [2] (вместо массивX и массивY нужно указать соответствующую область из таблицы с данными).

Получим  $r_{xy} = 0,9813$ . Можно сказать, что связь прямопропорциональная и довольно сильная.

Найдем остальные значения:

$$\bar{x} = \text{СРЗНАЧ}(\text{массивX}) = 5,5; \quad (3)$$

$$\bar{y} = \text{СРЗНАЧ}(\text{массивY}) = 14,7; \quad (4)$$

$$\sigma_x = \text{КОРЕНЬ}(\text{СРЗНАЧ}(\text{массивX}^2)) = 2,8723; \quad (5)$$

$$\sigma_y = \text{КОРЕНЬ}(\text{СРЗНАЧ}(\text{массивY}^2)) = 8,4976; \quad (6)$$

Следовательно,  $a = 2,903$ , а  $b = -1,267$ .

Т.о. вычислим  $y(11) = 30,667$ .

**Ответ:**  $y_x = 2,903x - 1,267$ ,  $y(11) = 30,667$ .

Мы решили данную задачу не прилагая особых усилий. За нас все вычисления выполнил компьютер. А если придется решать подобные задачи снова, то нам кроме замены таблицы с данными на новую таблицу, ничего делать не надо.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Гмурман В.Е. *Руководство к решению задач по теории вероятностей и математической статистике*. – М.: 1979г. - 400 с.  
 [2] Макарова Н.В., Трофинец В.Я. *Статистика в Excel*. – М.: 202г. - 368 с.

# Исследование двухлинейной системы массового обслуживания с одним местом в очереди

Небибуллаева Э. А.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

## 1. ПОСТАНОВКА ЗАДАЧИ

Рассматривается система массового обслуживания, состоящая из двух линий обслуживания, на которую поступает простейший поток заявок с интенсивностью  $\lambda$ .

Время обслуживания первой линией является непрерывная случайная величина  $\omega_1$  с интенсивностью  $\mu_1(x)$ .

Время обслуживания второй линией является экспоненциально распределенная случайная величина  $\omega_2$  с интенсивностью  $\mu_2$  ( $\mu_2 = const$ ).

Правила обслуживания состоят из следующих случаев:

Заявка, поступающая на свободную систему, начинает обслуживаться первой линией.

Если же одна из линий занята обслуживанием, то очередная заявка начинает обслуживаться оставшейся линией.

Если же обе линии заняты, то поступающая заявка остается в системе на месте для очереди.

Наконец, при занятых линиях и месте для очереди поступающая очередная заявка теряется.

## 2. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Система может находиться в следующих состояниях:

- (0)—система свободна от заявок;
- (1)—в системе находится одна заявка, обслуживаемая второй линией;
- (2)—в системе находится одна заявка, обслуживаемая первой линией;
- (3)—в системе находятся две заявки, обслуживаемые двумя линиями;
- (4)—в системе находятся две заявки, обслуживаемые двумя линиями, и одна — на месте ожидания.

Рассмотрим случайный процесс  $\xi(t)$ , фазовое пространство которого состоит из состояний системы.

Введем функции:

$$p_k(t) = P\{\xi(t) = k\}, \quad k = \overline{0, 4}$$

$$q_k(t, x) = P\{\xi(t) = k, \omega_1 < x\} \quad k = 2, 3, 4.$$

где  $P\{A\}$ —вероятность события  $A$ .

Заметим, что имеют место соотношения:

$$p_k(t) = \int_0^\infty q_k(t, x) dx, \quad k = 2, 3, 4.$$

При длительном наблюдении работа системы входит в так называемый стационарный режим, что означает эргодичность случайного процесса  $\xi(t)$ . Для функций, введенных для математического моделирования задачи существуют пределы при  $t \rightarrow +\infty$ .

Обозначим через:

$$p_k = \lim_{t \rightarrow +\infty} p_k(t), \quad k = \overline{0, 4}$$

$$g_k(x) = \lim_{t \rightarrow +\infty} q_k(t, x), \quad k = 2, 3, 4.$$

При переходе к пределу при  $t \rightarrow +\infty$  получаем следующую систему интегро-дифференциальных уравнений с граничными условиями:

$$-\lambda P_0 + \mu_2 P_1 + \int_0^\infty g_2(x) \mu_1(x) dx = 0 \quad (1)$$

$$-(\lambda + \mu_2) P_1 + \int_0^\infty g_3(x) \mu_1(x) dx = 0 \quad (2)$$

$$g_2'(x) + (\lambda + \mu_1(x)) g_2(x) = \mu_2 g_3(x) \quad (3)$$

$$g_3'(x) + (\lambda + \mu_1(x) + \mu_2) g_3(x) = \mu_2 g_4(x) + \lambda g_2(x) \quad (4)$$

$$g_4'(x) + (\mu_1(x) + \mu_2) g_4(x) = \lambda g_3(x) \quad (5)$$

$$g_2(0) = \lambda P_0 \quad (6)$$

$$g_3(0) = \lambda P_1 + \int_0^\infty g_4(x) \mu_1(x) dx \quad (7)$$

$$g_4(0) = 0 \quad (8)$$

Заметим, что в случае  $\mu_1(x) = const$  методика решения линейной однородной системы дифференциальных уравнений изложена в [3]

Для получения вероятностей состояний системы  $P_k$  можно составить алгебраическую систему условий стационарного равновесия:

$$\lambda P_0 = \mu_2 P_1 + \mu_1 P_2 \quad (9)$$

$$(\lambda + \mu_2) P_1 = \mu_1 P_3 \quad (10)$$

$$(\lambda + \mu_1) P_2 = \mu_2 P_3 + \lambda P_0 \quad (11)$$

$$(\lambda + \mu_1 + \mu_2) P_3 = \lambda P_1 + \lambda P_2 + (\mu_1 + \mu_2) P_4 \quad (12)$$

$$(\mu_1 + \mu_2)P_4 = \lambda P_3 \quad (13)$$

$$P_0 + P_1 + P_2 + P_3 + P_4 = 1 \quad (14)$$

Из которых можно выписать выражения для  $P_k$ ,  $k = \overline{0, 4}$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Анисимов В. В., Закусило О. К., Донченко В. С. *Элементы теории массового обслуживания и асимптотического анализа систем.* // Учебное пособие для вузов / Под общей редакцией В. В. Анисимова—1987
- [2] Гнеденко Б. В., Коваленко И. Н. *Введение в теорию массового обслуживания.*
- [3] Степанов В. В. *Курс дифференциальных уравнений*

## Применение нейронных сетей для задач управления

*Негримовская А. Я.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
 ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
 КАФЕДРА ДИФФЕРЕНЦИАЛЬНЫХ И ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ  
 (ГРУППА 500)

Вопросы управления динамическими объектами сегодня становятся все более актуальными. Одним из наиболее мощных инструментов создания интеллектуальных систем управления являются искусственные нейронные сети (ИНС), моделирующие базовые механизмы обработки информации в мозге. В работе была поставлена задача исследовать процесс управления нелинейной системы с помощью многослойной нейронной сети. Объектом исследования служит полносвязная многослойная нейронная сеть (МНС), обучаемая по методу обратного распространения ошибки ВР. Исследование условий устойчивого обучения сети основано на интерпретации процессов ее обучения как движений в аффинной системе, получаемой в результате эквивалентных преобразований исходной канонической структуры сети. МНС рассматривается как нелинейная динамическая система в расширенном фазовом пространстве. Отметим работы Комашинского В. И. и Смирнова Д. А., Махотило К. В., Терехова В. А. и Ефимова Д. В. А также статьи Беллюстина Н. С., Федяева О. И., Зубаня А. В. и Куликова С. А. наиболее близкие к данному направлению. Исследования применения искусственных нейронных сетей для

решения задач управления приводятся на основе дискретного описания динамической системы, рассматривается задача идентификации в теории автономного управления.

Рассмотрим применение искусственных нейронных сетей для решения задач управления. Введем векторы  $u(t) = (u_1(t), u_2(t), \dots, u_p(t))^T$ ,  $x(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$  и  $y(t) = (y_1(t), y_2(t), \dots, y_m(t))^T$ . Здесь  $u_i(t)$  и  $y_i(t)$  — вход и выход системы, соответственно,  $x_i(t)$  — переменная состояния системы,  $p$  — размерность входного пространства,  $m$  — размерность выходного пространства, а  $n$  — порядок системы. Классически, динамика такой системы описывается системой дифференциальных уравнений:

$$\begin{aligned} \frac{dx(t)}{dt} &= \Phi(x(t), u(t)); \\ y(t) &= F(x(t)). \end{aligned} \quad (1)$$

Здесь вектора функции  $\Phi = (\phi_1, \phi_2, \dots, \phi_n)$  и  $F = (f_1, f_2, \dots, f_m)$  — статические нелинейные преобразования.  $\Phi: \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ ,  $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ . Вектор  $x(t)$  описывает состояние системы в момент времени  $t$ . Он определяется состоянием системы в начальный момент  $t_0 < t$  и входом  $u$ , определенном на интервале  $[t_0, t)$ . Выход системы  $y(t)$  полностью определяется состоянием системы  $x$  в момент  $t$ .

Задача идентификации является фундаментальной в теории систем и, в частности, теории автоматического управления. Целью идентификации является построение идентификационной модели  $\hat{P}$ , аппроксимирующей объект  $P$ :

$$\|\hat{y} - y\| = \|\hat{P}(u) - P(u)\| \leq \varepsilon, \quad u \in U, \quad (2)$$

для некоторого заданного  $\varepsilon > 0$  и определенной нормы  $\|\cdot\|$ . Здесь  $\hat{y} = \hat{P}(u)$  — выход идентификационной модели,  $U$  — допустимое множество управления. Причем, как для статической, так и для динамической систем оператор  $P$  неявно определен парами вход-выход  $\{u, y\}$ .

Выбор класса, к которому принадлежит оператор  $\hat{P}$ , и самого оператора определяется множеством факторов, связанных с требуемой точностью и аналитической трактуемостью модели. К ним относятся адекватность представления  $P$  с помощью  $\hat{P}$ , сложность идентификации, простота модели, возможность ее расширения и дополнения, а также возможность использования модели в реальном масштабе времени. Выбор  $\hat{P}$  зависит и от имеющейся априорной информации о структуре объекта.

Целью обучения является получение закона управления, обеспечивающего заданное (обычно экстремальное) значение некоторого функционала качества

$I$ , определяемого текущим состоянием системы.

$$u(t) \Rightarrow I(y(t)) \rightarrow ext. \quad (3)$$

Алгоритм обратного распространения имеет существенные недостатки, поэтому в течение последних десяти лет в качестве альтернативных активно развиваются эволюционные методы синтеза нейросетевых систем управления (НСУ). В работе используется предложенная и апробированная методика синтеза НСУ на базе генетического алгоритма (ГА) [1]. Однако полученные результаты касались только случая линейного динамического объекта управления. Можно показать универсальность такой методики эволюционного синтеза НСУ как для линейных, так и нелинейных динамических объектов.

Тестовый нелинейный динамический объект управления описывается следующей системой:

$$T^2 x'' + 2T\xi(x)x' + x = ku, \quad (4)$$

где  $T = 0,5$ ,  $k = 1$ ;

$$\xi(x) = (|x| - 0,5)^2. \quad (5)$$

На рис. 1, 2 представлены переходные процессы в объекте для разных амплитуд скачкообразного сигнала задания. Как видно, из-за нелинейности объекта его динамические характеристики достаточно сильно меняются в зависимости от ве

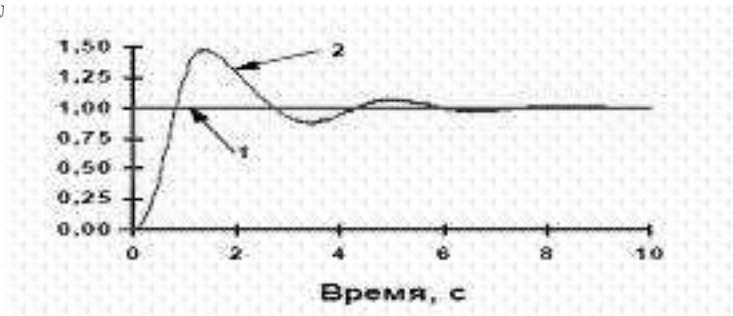


Рис. 1. Переходные характеристики объекта управления: кривая 1 — входной сигнал; кривая 2 — реакция НСУ; входной сигнал  $u = Al(t)$ ;  $A = 1$ .

Настройка параметров нейроконтроллера проводится с помощью основанного на генетическом алгоритме ГА метода стимулируемого контролируемого обучения. При этом используется тренировочный набор сигналов вида:

$$u_r^i(k) = A_i l(k), \quad A_i = -1,25 + 0,25i, \quad i = 1,9, \quad t \in [0, \tau], \quad (6)$$

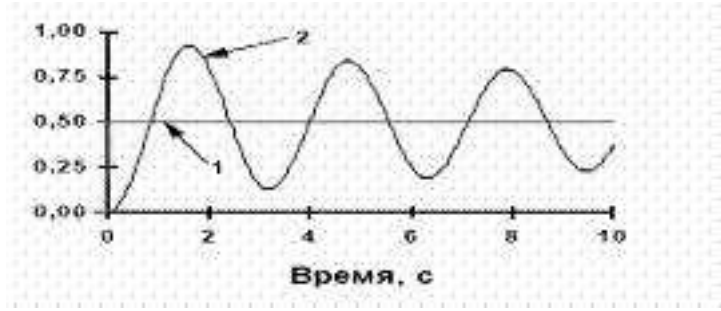


Рис. 2. Переходные характеристики объекта управления: кривая 1 — входной сигнал; кривая 2 — реакция НСУ; входной сигнал  $u = Al(t)$ ;  $A = 0,5$ .

где  $l(k)$  — дискретный аналог единичной функции Хевисайда. Длительность тренировочных входных сигналов равна  $t = 10$  с. Целью обучения является определение оптимальных значений синаптических параметров МСП, обеспечивающих минимум интегральной оценки качества работы НСУ при подаче на ее вход тренировочных сигналов. Использование скачкообразных тренировочных сигналов гарантирует синтез НСУ с малой статической ошибкой, однако переходная характеристика НСУ, обучаемой только по сигналам вида  $u_r = Al(t)$ , может быть сильно колебательной. Для того чтобы обеспечить системе необходимые динамические характеристики, включим в функционал качества кроме интегральной ошибки управления по выходному сигналу, также интегральную ошибку по его производной. Соответствующий функционал качества имеет вид:

$$I(y) = I_1(y) + \beta I_2(y), \quad (7)$$

где

$$I_1(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left( k \frac{y_i(k) - u_r^i(k)}{|A_i| + \varepsilon} \right), \quad (8)$$

$$I_2(y) = \frac{1}{M} \sum_{i=1}^M \frac{1}{k_\tau} \sum_{k=0}^{k_\tau} \left( k \frac{y_i(k) - y_i(k-1) - u_r^i(k) + u_r^i(k-1)}{(|A_i| + \varepsilon)\Delta} \right), \quad (9)$$

где  $M$  — размер тренировочного набора сигналов, равный 9;  $\varepsilon = 0,01$  — параметр, обеспечивающей непрерывность функционала качества;  $k_\tau = \tau/\Delta$  — количества шагов (длительность) работы НСУ. Величина веса  $\beta = 0,05$  выбрана такой, чтобы уравнять вклады в результирующий функционал  $I$  ошибок управления по выходу  $I_1$  и его производной  $I_2$ . Длительность тренировочных входных сигналов примем равной  $\tau = 10$  с.



Управление на основе многослойной нейронной сети, относится к интеллектуальным технологиям управления и обработки информации [2, 3, 4, 5, 6]. Эти технологии позволяют решать плохо формализуемые задачи управления сложными динамическими объектами в тех случаях, когда априорные «жесткие» модели и алгоритмы не адекватны реальному состоянию управляемого процесса.

Для анализа устойчивости процессов в многослойных нейронных сетях были использованы результаты по устойчивости положения равновесия и периодического решения систем обыкновенных дифференциальных уравнений. На основе этих результатов исследованы условия устойчивости процессов преобразования и обучения в статических многослойных нейронных сетях и получены оценки параметра  $y(t)$ , обеспечивающие устойчивость собственных и вынужденных движений в введенных эквивалентных структурах.

В качестве базового метода исследования устойчивости процессов обучения использовался метод функции Ляпунова. Использовали предположение о существовании единственного положения равновесия системы, в окрестности которого определены начальные условия.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Махотило К. В.* Разработка методик эволюционного синтеза нейросетевых компонентоа систем управления. — Диссертация на соискание ученой степени кандидата технических наук. [Электронный ресурс]. — Режим доступа: <[www.ai.xss.ru/docs/exponenta.doc](http://www.ai.xss.ru/docs/exponenta.doc)>.
- [2] *Сабанин В. Р.* Модифицированный генетический алгоритм для задач оптимизации в управлении [Электронный ресурс] / Сабанин В. Р., Смирнов Н. И., Репин А. И. — Режим доступа: <<http://www.neuroproject.ru/Papers/exponenta.doc>>.
- [3] *Тихонов А. Е.* Искусственные нейронные сети в задачах управления. [Электронный ресурс]. — Режим доступа: <<http://www.uran.donetsk.ua/masters/2004/fvti/tikhonov/diss/index.html>>.
- [4] *Садовой А. В.* Алгоритмы обучения нейронных сетей будущего [Электронный ресурс] / Садовой А. В., Сотник С. Л. — Режим доступа: <<http://prof9.narod.ru/doc/doc003.html>>.
- [5] *Камаев В. А.* Алгоритм определения оптимальной структуры нейронной сети в процессе адаптации [Электронный ресурс] / Камаев В. А., Щербаков М. В. — Режим доступа: <<http://www.dgma.donetsk.ua/ek/sc/neuro2002/index.html>>.
- [6] *Денисова Е. В.* Сравнительный анализ приближенных алгебраических и нейросетевых методов решения томографической задачи [Электронный ресурс] / Денисова Е. В., Денисов И. В., Кульчин Ю. Н. — Режим доступа: <<http://www.dgma.donetsk.ua/kiber/neuro2002/art22.html>>.

## О некоторых свойствах ядер Фредгольма специального типа

Османо́ва Э. А.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА АЛГЕБРЫ И ФУНКЦИОНАЛЬНОГО АНАЛИЗА (ГРУППА 500)

Теория  $*$ -алгебр - это одна из наиболее развитых и богатых приложениями составных частей современной математики. Она уже давно стала привычным аппаратом не только для математиков различных специальностей, но и для физиков-теоретиков.

Рассмотрим  $*$ -алгебру  $L_2(\mathcal{D})$ ,  $D = [a, b] \times [a, b]$  ядер Фредгольма с нормой

$$\| \mathcal{K} \|_D = \left( \int_a^b \int_a^b | \mathcal{K}(t, s) |^2 dt ds \right)^{\frac{1}{2}},$$

произведением

$$(\mathcal{K} \circ \mathcal{L})(t, s) = \int_a^b \mathcal{K}(t, u) \mathcal{L}(u, s) du$$

и инволюцией

$$\mathcal{K}^*(t, s) = \overline{\mathcal{K}(s, t)}.$$

Хорошо известна процедура, которая позволяет добавить единицу к нормированной  $*$ -алгебре без единицы [1]. В случае алгебры  $L_2(\mathcal{D})$  унитарное её  $*$ -расширение имеет вид  $\widetilde{L}_2(\mathcal{D}) = L_2(\mathcal{D}) \oplus \mathbb{C}$ . Операции сложения, умножения и инволюция задаются в алгебре  $\widetilde{L}_2(\mathcal{D})$  естественным образом:

$$\langle \mathcal{K}, \alpha \rangle + \langle \mathcal{L}, \beta \rangle = \langle \mathcal{K} + \mathcal{L}, \alpha + \beta \rangle, \langle \mathcal{K}, \alpha \rangle^* = \langle \mathcal{K}^*, \bar{\alpha} \rangle,$$

$$\langle \mathcal{K}, \alpha \rangle \otimes \langle \mathcal{L}, \beta \rangle = \langle \mathcal{K} \circ \mathcal{L} + \beta \mathcal{K} + \alpha \mathcal{L}, \alpha \beta \rangle.$$

Единица этой алгебры имеет вид  $\mathcal{E} = \langle \mathcal{O}, 1 \rangle$ , где  $\mathcal{O}$  - нулевое ядро.

**Определение 1.** Элемент  $\langle \mathcal{K}, \lambda \rangle \in \widetilde{L}_2(\mathcal{D})$  обратим, если существует элемент  $\langle \mathcal{L}, \mu \rangle \in \widetilde{L}_2(\mathcal{D})$ , такой что:

$$\langle \mathcal{K}, \lambda \rangle \otimes \langle \mathcal{L}, \mu \rangle = \langle \mathcal{L}, \mu \rangle \otimes \langle \mathcal{K}, \lambda \rangle = \mathcal{E}.$$

**Определение 2.** Элемент  $\langle \mathcal{K}, \lambda \rangle \in \widetilde{L}_2(\mathcal{D})$  называется унитарным, если

$$\langle \mathcal{K}, \lambda \rangle \otimes \langle \mathcal{K}^*, \bar{\lambda} \rangle = \mathcal{E}.$$

"Пересчет" этих определений на исходную алгебру приводит нас к определению квазиунитарного и квазиобратимого элементов.

**Определение 3.** Ядро Фредгольма  $\mathcal{K}$  называется квазиунитарным, если

$$\mathcal{K} \circ \mathcal{K}^* + \mathcal{K} + \mathcal{K}^* = \mathcal{O}.$$

**Определение 4.** Ядро Фредольма  $\mathcal{K}$  называется квазиобратимым, если существует  $\mathcal{L} \in L_2(\mathcal{D})$ , такое что

$$\mathcal{K} \circ \mathcal{L} + \mathcal{K} + \mathcal{L} = \mathcal{O}.$$

**Определение 5.** Ядра Фредольма  $\mathcal{K}$  и  $\mathcal{L}$  коммутативны, если

$$\mathcal{K} \circ \mathcal{L} = \mathcal{L} \circ \mathcal{K}.$$

**Определение 6.** Ядро Фредольма  $\mathcal{K}$  называется нормальным, если  $\mathcal{K} \circ \mathcal{K}^* = \mathcal{K}^* \circ \mathcal{K}$ .

Рассмотрим теперь множество ядер Фредгольма следующего вида:

$$\mathcal{K}(t, s) = \begin{cases} \alpha t + \beta s, & 0 \leq t \leq s \leq 1 \\ \gamma t + \delta s, & 0 \leq s \leq t \leq 1, \end{cases} \quad (1)$$

где  $\alpha, \beta, \gamma, \delta \in \mathbf{C}$

Сопряженное ядро вычисляется по формуле:

$$\mathcal{K}^*(t, s) = \begin{cases} \bar{\beta}t + \bar{\alpha}s, & 0 \leq s \leq t \leq 1 \\ \bar{\delta}t + \bar{\gamma}s, & 0 \leq t \leq s \leq 1. \end{cases}$$

Доказаны следующие предложения.

**Теорема 1.** Два ядра Фредгольма:  $\mathcal{K}$  вида (1) и  $\mathcal{L}(t, s) = \begin{cases} \varepsilon t + \xi s, & 0 \leq t \leq s \leq 1 \\ \eta t + \tau s, & 0 \leq s \leq t \leq 1 \end{cases}$  коммутативны тогда и только тогда, когда выполняются условия :

$$\gamma\varepsilon = \alpha\eta, \quad \beta\tau = \delta\xi, \quad \delta\varepsilon = \alpha\tau, \quad \gamma\xi = \beta\eta, \quad \alpha\xi = \beta\varepsilon, \quad \gamma\tau = \delta\eta.$$

**Следствие 1.** Для коммутруемости ядер  $\mathcal{K}$  и  $\mathcal{L}$  необходимо, чтобы коммутировали матрицы  $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$  и  $\begin{pmatrix} \varepsilon & \xi \\ \eta & \tau \end{pmatrix}$ .

**Замечание 1.** Указанное условие коммутруемости матриц не является достаточным. Для этого достаточно рассмотреть ядро

$\mathcal{R}(t, s) = \begin{cases} t, & 0 \leq t \leq s \leq 1 \\ s, & 0 \leq s \leq t \leq 1 \end{cases}$ , которое, очевидно, не коммутирует с любым ядром вида (1).

**Следствие 2.** Ядро Фредгольма  $\mathcal{K}$  вида (1) является нормальным тогда и только тогда, когда выполняются условия :

$$\alpha\bar{\delta} = \gamma\bar{\beta}, \quad \delta\bar{\beta} = \alpha\bar{\gamma}, \quad |\beta| = |\alpha|, \quad |\delta| = |\gamma|.$$

**Следствие 3.** Для нормальности ядра  $\mathcal{K}$  вида (1) необходимо, чтобы матрица  $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$  была нормальной.

**Замечание 2.** Условие нормальности указанной матрицы не является достаточным. Например, матрица  $\begin{pmatrix} 1 & i \\ i & 1-i \end{pmatrix}$  нормальная, а соответствующее ей ядро вида (1) не является нормальным.

В дальнейшем предполагается получить критерии квазиобратимости и квазиунитарности ядер вида (1).

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Рудин У. *Функциональный анализ*. — М., Мир, 1975. 437 с.

## Спектральные задачи, порождённые проблемами малых движений вязкоупругих и релаксирующих сред

Сёмкина Е. В.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА МАТЕМАТИЧЕСКОГО АНАЛИЗА (ГРУППА 500)

Начально-краевые задачи о малых движениях жидкости в полностью заполненном контейнере приводят к линейным интегродифференциальным уравнениям в гильбертовом пространстве  $\mathcal{H}$  вида

$$\frac{du}{dt} + A_0 u + \sum_{k=1}^m \int_0^t e^{-\gamma_k(t-s)} A_k u(s) ds = f(t), \quad u(0) = u^0 \quad (1)$$

для вязкоупругой жидкости, и

$$\frac{d^2 u}{dt^2} + A_0 u + \sum_{k=1}^m \int_0^t e^{-\gamma_k(t-s)} A_k u(s) ds = f(t), \quad u(0) = u^0, \quad u'(0) = u^1, \quad (2)$$

для идеальной релаксирующей жидкости.

В данной работе исследуются спектральные задачи ассоциированные с (1), (2). Точнее частный случай таких задач, когда выполнено условие

$$A_k = \alpha_k A_0^\delta, \quad \alpha_k > 0, \quad 0 < \delta < 1. \quad (3)$$

Приведём сначала известные общие результаты.

В работе [1] исследована задача Коши (1). Для этого уравнения осуществляется переход к дифференциальному уравнению в ортогональной сумме пространств

$$\frac{d\tilde{u}}{dt} + \mathcal{A}\tilde{u} = \tilde{f}(t), \quad \tilde{u}(0) = \tilde{u}^0$$

$$\begin{aligned}\tilde{\mathcal{H}} &:= \mathcal{H}_0 \oplus \widehat{\mathcal{H}}_1, \quad \mathcal{H}_0 := \mathcal{H}, \quad \widehat{\mathcal{H}}_1 := \bigoplus_{k=1}^m \mathcal{H}_k, \quad \mathcal{H}_k := \mathcal{H}, \quad k = 1, \dots, m. \\ \tilde{u}(t) &:= (u_0(t); \widehat{u}_1(t))^t, \quad \widehat{u}_1(t) := (u_1(t); \dots; u_m(t))^t.\end{aligned}$$

Исследована также ассоциированная спектральная задача

$$\mathcal{A}\tilde{u} = \lambda\tilde{u}, \quad \tilde{u} \in \mathcal{D}(\mathcal{A}), \quad (4)$$

где оператор  $\mathcal{A}$  задан на области определения

$$\mathcal{D}(\mathcal{A}) = \{\tilde{u} = (u_0; \widehat{u}_1)^\tau \in \tilde{H} : u_0 + A^{-1/2}Q_{10}^*\widehat{u}_1 \in \mathcal{D}(A_0)\}$$

посредством формулы

$$\mathcal{A}\tilde{u} = \begin{pmatrix} A_0(u_0 + A_0^{-1/2}Q_{10}^*\widehat{u}_1) \\ -Q_{10}A_0^{1/2}u_0 + A_{11}\widehat{u}_1 \end{pmatrix}, \quad \tilde{u} \in \mathcal{D}(\mathcal{A}),$$

$$\begin{aligned}Q_{10} &:= (Q_1; \dots; Q_m)^t, \quad Q_k := A_k^{1/2}A_0^{-1/2}, \quad \mathcal{D}(Q_{10}) = \mathcal{H}_0, \\ A_{11} &:= \text{diag}(\gamma_k I)_{k=1}^m.\end{aligned}$$

С использованием теории пространств с индефинитной метрикой для задачи (4), получены следующие результаты:

**Теорема 1.** *Спектр задачи (4) положителен, за исключением не более конечного числа (с учётом кратности) не вещественных собственных значений.*

**Теорема 2.** *Задача (4) имеет счётное множество положительных собственных значений  $\{\lambda_n^{(\infty)}\}_{n=1}^\infty$  с единственной предельной точкой  $\lambda = +\infty$  и собственными элементами  $\{\tilde{u}_n^{(\infty)}\}_{n=1}^\infty$ ,  $\tilde{u}_n^{(\infty)} = (u_{0n}^{(\infty)}; \widehat{u}_{1n}^{(\infty)})^\tau \in \tilde{H}$ , проекции  $\{u_{0n}^{(\infty)}\}_{n=1}^\infty$  которых на  $\mathcal{H}$  образуют базис Рисса с конечным дефектом в пространстве  $\mathcal{H}$ . Если выполнено условие*

$$A_0^{-1} \in \mathfrak{S}_{p_0},$$

*то указанный базис Рисса является  $p_0$ -базисом (с конечным дефектом) в  $\mathcal{H}$ .*

Рассмотрен также важный частный случай, когда операторы  $A_k$ ,  $k = \overline{1, m}$ , образующие матричный оператор  $\mathcal{A}$ , удовлетворяют условиям

$$A_k = \alpha_k A_0, \quad \alpha_k > 0 \quad k = \overline{1, m}. \quad (5)$$

Доказана соответствующая

**Теорема 3.** *Если выполнены условия (5), то спектр оператора  $\mathcal{A}$  дискретен, за исключением  $+\infty$  и  $m$  конечных точек сгущения  $\{\beta_k\}_{k=1}^m$ , являющихся нулями функции*

$$e_0(\lambda) := 1 + \sum_{k=1}^m \alpha_k / (\gamma_k - \lambda)$$

и удовлетворяющих условиям

$$0 < \gamma_1 < \beta_1 < \dots < \gamma_m < \beta_m < \infty.$$

При этом  $\sigma(\mathcal{A}) \setminus (\{\infty\} \cup \{\beta_k\}_{k=1}^m)$  состоит из не более, чем конечного множества не вещественных собственных значений и  $m+1$  ветви положительных собственных значений, причём одна из ветвей стремится к  $+\infty$ , а другие  $m - \kappa$   $\{\beta_k\}_{k=1}^m = \sigma_{ess}(\mathcal{A}) \setminus \{+\infty\}$ . Более того, если  $\{\tilde{u}_n\}_{n=1}^\infty$ ,  $u_n = (u_{0n}; \hat{u}_{1n})^\tau \in \hat{\mathcal{H}}$ , – собственные элементы оператора  $\mathcal{A}$ , отвечающие ветви собственных значений, стремящейся к  $+\infty$  ( $\kappa$  ветвям, стремящимся к  $\{\beta_k\}_{k=1}^m$ ), то их проекции  $\{u_{0n}^{(\infty)}\}_{n=1}^\infty$  на  $\mathcal{H}$  образуют базис Рисса с конечным дефектом в пространстве  $\mathcal{H}$  (соответственно, проекции  $\{u_{1n}^{(\infty)}\}_{n=1}^\infty$  на  $\hat{\mathcal{H}}_1$  образуют базис Рисса с конечным дефектом в пространстве  $\hat{\mathcal{H}}_1$ ). Если выполнено условие  $A_0^{-1} \in \mathfrak{S}_{p_0}$ , то указанные базисы Рисса являются  $p_0$ -базисами (с конечными дефектами) в  $\mathcal{H}$  и  $\hat{\mathcal{H}}_1$  соответственно.

Теорема о сильной разрешимости задачи Коши для уравнения (2) доказана в работе [2] по изложенной выше схеме (см. [1]). Ассоциированная спектральная задача

$$\hat{\mathcal{A}}y = \lambda \hat{\mathcal{J}}y, \quad y \in \mathcal{D}(\hat{\mathcal{A}}) \quad (6)$$

исследуется в статье [3]. Здесь

$$\hat{\mathcal{H}} := \mathcal{H} \oplus \left( \bigoplus_{k=1}^m \mathcal{H}_k \right) \oplus \mathcal{H}, \quad \mathcal{H}_k := \mathcal{H},$$

$$\hat{\mathcal{A}} = \begin{pmatrix} A_0 & A_0^{1/2}(\hat{Q}^\tau)^* & 0 \\ \hat{Q}^\tau A_0^{1/2} & -\hat{\gamma} & 0 \\ 0 & 0 & -I \end{pmatrix},$$

$$\hat{Q} := (A_1^{1/2}A_0^{-1/2}; \dots; A_m^{1/2}A_0^{-1/2}) \quad \hat{\gamma} := \text{diag}(\gamma_1 I, \dots, \gamma_m I),$$

$$\mathcal{D}(\hat{\mathcal{A}}) = \{(u; \hat{v}; z)^\tau \in \hat{\mathcal{H}} \mid u + A_0^{-1/2}(\hat{Q}^\tau)^* \hat{v} \in \mathcal{D}(A_0)\}.$$

Доказаны теоремы о том, что соответствующий этой задаче пучок

$$L(\lambda) = I + \lambda^2 A_0^{-1} - \sum_{k=1}^m \frac{1}{\lambda - \gamma_k} B_k, \quad B_k := (A_k^{1/2} A_0^{-1/2})^* (A_k^{1/2} A_0^{-1/2})$$

имеет  $m$  ветвей собственных значений с предельными точками  $\beta_1, \dots, \beta_m$  и эти ветви между собой не пересекаются, а также две комплексно сопряжённые ветви собственных значений, локализованные в области

$$\Lambda := \{\lambda \in \mathbb{C} \mid - \sum_{k=1}^m \|B_k\| < \text{Re} \lambda < 0, \text{Im} \lambda \neq 0\}.$$

Собственные элементы  $L(\lambda)$ , отвечающие каждой вещественной ветви собственных значений, образуют базис Рисса в гильбертовом пространстве  $\mathcal{H}$ , а система собственных и присоединённых элементов, отвечающая комплексным собственным значениям, образует полную в  $\mathcal{H}^2$  систему при выполнении некоторых дополнительных условий.

В нашем случае, когда выполнено условие (3), спектр задач (4) и (6) качественно изменяется, а именно:

1. Для задачи (4) характеристическое уравнение примет вид

$$\sum_{k=1}^m \frac{\alpha_k}{\gamma_k - \lambda} = \lambda(\lambda_n(A_0))^{-\delta} - (\lambda_n(A_0))^{1-\delta}, \quad n \in \mathbb{N}.$$

Предельными точками дискретного вещественного спектра здесь будут  $\{\gamma_k\}_{k=1}^m$  и  $+\infty$ . В этом случае в точности выполняется теорема 3, если заменить везде  $\beta_k$  на  $\gamma_k$ .

2. Для задачи (6) запишем соответствующее характеристическое уравнение

$$\sum_{k=1}^m \frac{\alpha_k}{\lambda - \gamma_k} = \lambda^2(\lambda_n(A_0))^{-\delta} + (\lambda_n(A_0))^{1-\delta}, \quad n \in \mathbb{N}.$$

Здесь также предельными точками дискретного вещественного спектра будут  $\{\gamma_k\}_{k=1}^m$ . И все результаты для задачи (6) сохраняются в заменой  $\beta_k$  на  $\gamma_k$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Т.Я.Азизов, Н.Д.Копачевский, Л.Д.Орлова *Эволюционная и спектральная задачи, порождённые проблемой малых движений вязкоупругой жидкости.*//Труды Санкт-Петербургского математического общества. – Т. 6, – 1998. – С.5-33.
- [2] Н.Д.Копачевский *Задача Коши для линейного интегродифференциального уравнения второго порядка в гильбертовом пространстве* // Ученые записки Таврического национального университета им. В.И. Вернадского. – Сер. «Математика. Механика. Информатика и кибернетика». – 2003. – Т. 16(55), № 1. – С. 139-152.
- [3] Д.А.Загора, Н.Д.Копачевский *О спектральной задаче, связанной с интегродифференциальным уравнением второго порядка.*//Ученые записки Таврического университета им. В.И. Вернадского. – Сер. «Математика. Механика. Информатика и кибернетика». – 2004. – Т.17(56), №2. С.11-29.

## Анализ на временных шкалах

*Сулейманова А. Р.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА АЛГЕБРЫ И ФУНКЦИОНАЛЬНОГО АНАЛИЗА (ГРУППА 401)

*Временная шкала* (которая является специальным случаем *измеримой цепи*, рассм. в гл. 8) является произвольным непустым замкнутым подмножеством действительных чисел. Таким образом,

$$\mathbb{R}, \quad \mathbb{Z}, \quad \mathbb{N}, \quad \mathbb{N}_0,$$

являются примерами временных шкал, так же, как

$$[0, 1] \cup [2, 3], \quad [0, 1] \cup \mathbb{N}, \quad \text{канторово множество},$$

в то время, как

$$\mathbb{Q}, \quad \mathbb{R} \setminus \mathbb{Q}, \quad \mathbb{C}, \quad (0, 1),$$

и открытый интервал между 0 и 1, *не* временные шкалы. Обозначим временную шкалу символом  $\mathbb{T}$ .

Цель работы – основное определение Временной шкалы  $\mathbb{T}$ , дифференцирование.

Анализ временных шкал был начат Stefan Hilger в его PhD тезисах [159] для создания теории, которая может объединить дискретный и непрерывный анализ. Действительно, ниже введем дельта-производную  $f^\Delta$  для функции  $f$ , определенной на  $\mathbb{T}$ , причём

(i)  $f^\Delta = f'$  является обычной производной, если  $\mathbb{T} = \mathbb{R}$ ,

(ii)  $f^\Delta = \Delta f$  является обычным передним разностным оператором, если  $\mathbb{T} = \mathbb{Z}$ .

**Определение 1.1.** Пусть  $\mathbb{T}$  – временная шкала. Для  $t \in \mathbb{T}$  определим оператор переднего скачка  $\sigma : \mathbb{T} \rightarrow \mathbb{T}$  равенством:

$$\sigma(t) := \inf\{s \in \mathbb{T} : s > t\},$$

и оператор заднего скачка  $\rho : \mathbb{T} \rightarrow \mathbb{T}$  равенством:

$$\rho(t) := \sup\{s \in \mathbb{T} : s < t\}.$$

В это определение  $\inf \emptyset = \sup \mathbb{T}$  (т.е.  $\sigma(t) = t$ , если  $\mathbb{T}$  имеет максимум  $t$ ) и  $\sup \emptyset = \inf \mathbb{T}$  (т.е.  $\rho(t) = t$ , если  $\mathbb{T}$  имеет минимум  $t$ ), где  $\emptyset$  – пустое множество. Если  $\sigma(t) > t$ , скажем, что точка *t-рассеянная справа*, если же  $\rho(t) < t$ , то скажем, что *t-рассеянная слева*. Точки, одновременно рассеянные и справа, и слева, называются *изолированными*.

**Табл.1. Классификация точек.**



|                         |                           |
|-------------------------|---------------------------|
| $t$ – рассеянная справа | $t < \sigma(t)$           |
| $t$ – плотная справа    | $t = \sigma(t)$           |
| $t$ – рассеянная слева  | $\rho(t) < t$             |
| $t$ – плотная слева     | $\rho(t) = t$             |
| $t$ – изолированная     | $\rho(t) < t < \sigma(t)$ |
| $t$ – плотная           | $\rho(t) = t = \sigma(t)$ |

Также, если  $t < \sup(\mathbb{T})$  и  $\sigma(t) = t$ , то точку  $t$  назовём *плотной справа*, а если  $t > \inf(\mathbb{T})$  и  $\rho(t) = t$ , то точку  $t$  назовём *плотной слева*. Точки, плотные справа и плотные слева одновременно, назовём плотными. Наконец, *функция зернистости*  $\mu : \mathbb{T} \rightarrow [0, \infty)$  определяется как

$$\mu(t) := \sigma(t) - t.$$

**Пример 1.1.** Кратко рассмотрим два примера:  $\mathbb{T} = \mathbb{R}$  и  $\mathbb{T} = \mathbb{Z}$ .

(i) Если  $\mathbb{T} = \mathbb{R}$ , то для каждого  $t \in \mathbb{R}$

$$\sigma(t) = \inf\{s \in \mathbb{R} : s > t\} = \inf(t, \infty) = t,$$

и аналогично  $\rho(t) = t$ . Следовательно, каждая точка  $t \in \mathbb{R}$  является плотной. Отсюда следует, что функция зернистости  $\mu$  имеет вид

$$\mu(t) \equiv 0 \quad \text{для всех } t \in \mathbb{T}.$$

(ii) Если  $\mathbb{T} = \mathbb{Z}$ , то для каждого  $t \in \mathbb{Z}$

$$\sigma(t) = \inf\{s \in \mathbb{Z} : s > t\} = \inf\{t+1, t+2, t+3, \dots\} = t+1,$$

и аналогично  $\rho(t) = t-1$ . Следовательно, каждая точка  $t \in \mathbb{Z}$  является изолированной. Функция зернистости  $\mu$  в этом случае имеет вид

$$\mu(t) \equiv 1 \quad \text{для всех } t \in \mathbb{T}.$$

Если  $\mathbb{T} = \mathbb{R}$ , тогда получаем хорошо известное дифференциальное уравнение Риккати

$$z' + q(t) + \frac{1}{p(y)}z^2 = 0,$$

а если  $\mathbb{T} = \mathbb{Z}$ , то получаем разностное уравнение Риккати

$$\Delta z_k + q(t) + \frac{z_k^2}{p(t) + z_k} = 0.$$

**Упражнение 1.1.** Для каждой из следующих временных шкал  $\mathbb{T}$ , найти  $\sigma$ ,  $\rho$ , и  $\mu$ , и классифицировать каждую точку  $t \in \mathbb{T}$  как плотную слева, рассеянную слева, плотную справа, рассеянную справа:

- (i)  $\mathbb{T} = \{2^n : n \in \mathbb{Z}\} \cup \{0\}$ ;  
(ii)  $\mathbb{T} = \{\frac{1}{n} : n \in \mathbb{N}\} \cup \{0\}$ ;

*Решение.*

- (i)  $\mathbb{T} = \{2^n : n \in \mathbb{Z}\} \cup \{0\}$ ;  
 $\mathbb{T} = \{0, \dots, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8, \dots\}$ ,  $t \in \mathbb{T}, t \neq 0$ ;  
 $\sigma(\mathbb{T}) = \inf\{s \in \mathbb{T} : s > t\} = \inf\{2t, 4t, 8t, \dots\} = 2t$ ;  
 $\rho(\mathbb{T}) = \sup\{s \in \mathbb{T} : s < t\} = \sup\{\frac{t}{2}, \frac{t}{4}, \frac{t}{8}, \dots\} = \frac{t}{2}$ ;  
 $\mu(t) = \sigma(t) - t = t$ ;  
 $\rho(t) < t < \sigma(t) \Rightarrow t$  — изолированная точка;  
 $\sigma(0) = \inf\{s \in \mathbb{T} : s > 0\} = 0$ ;  
 $\rho(0) = \sup\{s \in \mathbb{T} : s < 0\} = \sup\{\emptyset\} = 0$ ;  
 $\mu(0) = 0$ .

- (ii)  $\mathbb{T} = \{\frac{1}{n} : n \in \mathbb{N}\} \cup \{0\}$ ;  
 $\mathbb{T} = \{0, \dots, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\}$ ,  $t \in \mathbb{T}, t = \frac{1}{k}$ ;  
 $\sigma(\mathbb{T}) = \inf\{s \in \mathbb{T} : s > t\} = \inf\{\frac{1}{k-1}, \frac{1}{k-2}, \dots, 1\} = \frac{1}{k-1}$ ;  
 $\rho(\mathbb{T}) = \sup\{s \in \mathbb{T} : s < t\} = \sup\{0, \dots, \frac{1}{k+3}, \frac{1}{k+2}, \frac{1}{k+1}\} = \frac{1}{k+1}$ ;  
 $\mu(t) = \sigma(t) - t = \frac{1}{k-1} - \frac{1}{k} = \frac{1}{k(k-1)}$ ;  
 $\rho(t) < t < \sigma(t) \Rightarrow t$  — изолированная точка.

**Упражнение 1.2.** Является ли отображение  $\sigma : \mathbb{T} \rightarrow \mathbb{T}$  взаимно — однозначным? Отображением "на"? Если оно не является отображением "на" определить область изменения  $\sigma(t)$  отображения  $\sigma$ . Рассмотрим также  $\rho : \mathbb{T} \rightarrow \mathbb{T}$ ? Это утверждение было предложено Романом Хильшером.

*Решение.*

Рассмотрим функцию  $\sigma(t)$ .

1)  $\sigma(t)$  — не является взаимно-однозначным.

Действительно, пусть  $\mathbb{T} = (-\infty, 0] \cup [1, +\infty)$ . Рассмотрим точки  $t_1 = 1$  и  $t_2 = 0$ ,  $t_1 \neq t_2$ .

$$\sigma(t_1) = \sigma(1) = \inf\{s \in \mathbb{T} : s > 1\} = 1,$$

$$\sigma(t_2) = \sigma(0) = \inf\{s \in \mathbb{T} : s > 0\} = 1,$$

Т.е. получаем, что для  $t_1 \neq t_2$ ,  $\sigma(t_1) = \sigma(t_2)$ . Что и требовалось доказать.

2)  $\sigma(t)$  — не является отображением "на".

Действительно, пусть временная шкала имеет вид  $\mathbb{T} = \{t_1\} \cup [t_2, +\infty)$ , где  $t_1 < t_2$ .

В данном случае  $\sigma(t_1) = \sigma(t_2) = t_2$ , то  $\sigma(\mathbb{T}) = [t_2, +\infty) \neq \mathbb{T}$ . Что и требовалось доказать.

Определим область изменения  $\sigma(t)$  отображения  $\sigma$ :

- а)  $\sigma(\mathbb{T}) \subset \mathbb{T}$ ;

- b)  $\sigma(\mathbb{T})$  – замкнуто;
- c)  $\sup \sigma(\mathbb{T}) = \sup \mathbb{T}$ .
- d)  $t$ – рассеянная справа  $\Leftrightarrow t \in \overline{\sigma(\mathbb{T})}$ .

Рассмотрим функцию  $\rho(t)$ .

- 1)  $\rho(t)$  – не является взаимно-однозначным.

Действительно, пусть  $\mathbb{T} = (-\infty, 0] \cup [1, +\infty)$ . Рассмотрим точки  $t_1 = 1$  и  $t_2 = 0$ ,  $t_1 \neq t_2$ .

$$\rho(t_1) = \rho(1) = \sup\{s \in \mathbb{T} : s < 1\} = 0,$$

$$\rho(t_2) = \rho(0) = \sup\{s \in \mathbb{T} : s < 0\} = 0,$$

Т.е. получаем, что для  $t_1 \neq t_2$ ,  $\rho(t_1) = \rho(t_2)$ . Что и требовалось доказать.

- 2)  $\rho(t)$  – не является отображением "на".

Действительно, пусть временная шкала имеет вид  $\mathbb{T} = (-\infty, t_1] \cup \{t_2\}$ , где  $t_1 < t_2$ .

В данном случае  $\rho(t_1) = \rho(t_2) = t_1$ , то  $\rho(\mathbb{T}) = [t_2, +\infty) \neq \mathbb{T}$ . Что и требовалось доказать.

Определим область изменения  $\rho(t)$  отображения  $\rho$ :

- a)  $\rho(\mathbb{T}) \subset \mathbb{T}$ ;
- b)  $\rho(\mathbb{T})$  – замкнуто;
- c)  $\inf \rho(\mathbb{T}) = \inf \mathbb{T}$ .
- d)  $t$ – рассеянная слева  $\Leftrightarrow t \in \overline{\rho(\mathbb{T})}$ .

**Упражнение 1.3.** Если  $\mathbb{T}$  состоит из конечного числа точек, вычислить

$$\sum_{t \in \mathbb{T}} \mu(t).$$

*Решение.*

Рассмотрим временную шкалу, состоящую из конечного числа точек

$$\mathbb{T} = \{t_1, t_2, t_3, \dots, t_n\}.$$

$$\text{То } \sigma(t_i) = \inf\{s \in \mathbb{T} : s > t_i\} = \inf\{t_{i+1}, t_{i+2}, t_{i+3}, \dots, t_n\} = t_{i+1}, \quad i = \overline{1, n-1}.$$

Следовательно,  $\mu(t_i) = \sigma(t_i) - t_i = t_{i+1} - t_i, \quad i = \overline{1, n-1}$ .

$$\mu(t_n) = \sigma(t_n) - t_n = t_n - t_n = 0$$

Просуммируем

$$\begin{aligned} \sum_{t \in \mathbb{T}} \mu(t) &= \sum_{k=1}^{n-1} (t_{k+1} - t_k) = t_2 - t_1 + t_3 - t_2 + t_4 - t_3 + \dots + t_n - t_{n-1} = \\ &= t_n - t_1. \text{ Для } i = n. \end{aligned}$$

Теперь рассмотрим функцию  $f : \mathbb{T} \rightarrow \mathbb{R}$  и определяем так называемую *дельта производную* функции  $f$  в точке  $t \in \mathbb{T}^k$ .

**Определение 1.2.** Предположим, что  $f : \mathbb{T} \rightarrow \mathbb{R}$  – функция и пусть  $t \in \mathbb{T}^k$ . Тогда определяем  $f \Delta(t)$  как число (при условии существования) со свойством, что для любого фиксированного  $\varepsilon > 0$ , существует окрестность  $U$

в точке  $t$  (т.е.  $U = (t - \delta, t + \delta) \cap \mathbb{T}$  для некоторого  $\delta > 0$ ) такая, что

$$|[f(\sigma(t)) - f(s)] - f^\Delta(t)[\sigma(t) - s]| \leq \varepsilon |\sigma(t) - s| \quad \forall s \in U$$

Назовем  $f^\Delta(t)$  — *дельта производной* функции  $f$  в точке  $t$ .

**Пример 1.2.** (i) Если  $f : \mathbb{T} \rightarrow \mathbb{R}$  определена как  $f(t) = \alpha$  для всех  $t \in \mathbb{T}$ , где  $\alpha \in \mathbb{R}$  — константа, то  $f^\Delta(t) \equiv 0$ . Это ясно, т.к. для любого  $\varepsilon > 0$ ,

$$|f(\sigma(t)) - f(s) - 0 \cdot [\sigma(t) - s]| = |\alpha - \alpha| = 0 \leq \varepsilon \cdot |\sigma(t) - s|$$

выполняется для всех  $s \in \mathbb{T}$ .

(ii) Если  $f : \mathbb{T} \rightarrow \mathbb{R}$ , определена как  $f(t) = t$  для всех  $t \in \mathbb{T}$ , то  $f^\Delta(t) \equiv 1$ . Это следует из того, что для любого  $\varepsilon > 0$ ,

$$|f(\sigma(t)) - f(s) - 1 \cdot [\sigma(t) - s]| = |\sigma(t) - s - (\sigma(t) - s)| = 0 \leq \varepsilon \cdot |\sigma(t) - s|$$

выполняется для всех  $s \in \mathbb{T}$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Martin Bohner, Allan Peterson // Dynamic Equations on Time Scales, PRELIMINARY FINAL Version from May 4, 2001. 355 с.

## Нелинейное интегральное уравнение Урысона первого рода

*Хазова Ю. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ДИФФЕРЕНЦИАЛЬНЫХ И ИНТЕГРАЛЬНЫХ УРАВНЕНИЙ (ГРУППА 403)

Уравнение Урысона кроме самостоятельного математического интереса имеет широкие приложения в эконометрике, в кинетической теории газов, в механике, в теории популяции, в прикладных задачах синтеза магнитного поля на оси катушки ЯМР-томографа, в нелинейных задачах синтеза излучающих систем, задачах дистанционного зондирования, задачах о волнах на поверхности движущейся жидкости.

Название оператора Урысона связано с именем Павла Самуиловича Урысона, который был учеником Николая Николаевича Лузина. П.С.Урысон написал первую в России работу по теории нелинейных операторов, относящуюся к еще не родившемуся функциональному анализу.

Оператором Урысона называется интегральный оператор вида [1]:

$$Az(t) = \int_a^b k[t, s, z(s)]ds. \quad (1)$$

В работе будем рассматривать уравнение Урысона первого рода:

$$Az \equiv \int_a^b n(s, t - z(s))ds = u(t), \quad c \leq t \leq d,$$

$z \in Z$ ,  $u \in U$ , где  $Z$  и  $U$  - некоторые метрические пространства,  $u(t)$  - известная функция.

Возьмем ядро частного вида:

$$n(s, t) = a(s)k(t),$$

где  $a(s)$  - заданная функция.

Уравнение (2) примет вид:

$$Az \equiv \int_a^b a(s)k(t - z(s))ds = u(t), \quad c \leq t \leq d. \quad (3)$$

Рассмотрим несколько характерных случаев. Предположение о монотонности искомой функции. Пусть  $z(s)$  - монотонно возрастающая функция. Сделаем замену  $z(s) = \tau$  (существует обратная функция  $s = \varphi(\tau)$ ,  $ds = \varphi'(\tau)d\tau$ ), получаем интегральное уравнение типа свертки первого рода:

$$B\psi \equiv \int_a^b \psi(\tau)k(t - \tau)d\tau = u(t), \quad c \leq t \leq d \quad (4)$$

относительно неизвестной функции  $\psi(\tau)$ . Для функции  $\varphi(\tau)$  получаем уравнение

$$\psi(\tau) = a(\varphi(\tau))\varphi'(\tau). \quad (5)$$

Задача (4) является некорректной, для ее решения применим метод регуляризации А.Н.Тихонова ([2], [3]) в предположении, что  $\psi \in W_2^1[a, b]$ :

$$M^\alpha[\psi] \equiv \alpha \left( \int_a^b [\psi^2(\tau) + \psi'^2(\tau)] d\tau \right) + \int_c^d \left( \int_a^b \psi(\tau)k(t-\tau)d\tau - u_\delta(t) \right)^2 dt, \quad (6)$$

$$\eta = (h, \delta), \quad \|B_h - B\| \leq h, \quad \|u_\delta - u\| \leq \delta.$$

Элемент  $\psi_\eta^\alpha$  является минимизирующим для  $M^\alpha[\psi]$ . Если решение (4) единственно и параметр регуляризации  $\alpha$  определенным образом согласован с погрешностью задания входной информации  $\delta$ :  $\alpha = \alpha(\delta)$ , то  $\psi_\eta^{\alpha(\delta)} \rightarrow \psi(\tau)$  при  $\delta \rightarrow 0$ .

В случае неединственности последовательность экстремалей  $\psi_\eta^{\alpha(\delta)}$  сходится к решению  $\psi(\tau)$  с минимальной нормой.

Решение исходного интегрального уравнения находится из решения обыкновенного дифференциального уравнения (5).

Применительно к уравнению (3) метод регуляризации А.Н.Тихонова приводит к регуляризованному нелинейному интегральному уравнению в форме:

$$\alpha z_\alpha(t) + \int_a^b m(\tau, t, z_\alpha(\tau), z_\alpha(t)) d\tau = f(t, z_\alpha(t)), \quad a \leq t \leq b, \quad (7)$$

$$m(\tau, t, z(\tau), z(t)) = \int_c^d n(\tau, s - z(\tau)) n_s(t, s - z(t)) ds$$

$$f(t, z(t)) = \int_c^d n_s(t, s - z(t)) u(s) ds,$$

где  $\alpha$  - параметр регуляризации.

Таким образом уравнение первого рода сводится к уравнению второго рода (7), для решения которых используются метод последовательных приближений, метод Ньютона-Канторовича, метод квадратур.

Во многих приложениях ядро является  $\delta$ -образной функцией [4].

Уравнение (3) на вещественной оси с ядром в виде обобщенной  $\delta$ -функции  $\delta(t)$  в случае монотонности функции  $z(s)$  соответствует уравнению:

$$\int_{-\infty}^{\infty} \delta(t-\tau) \varphi(\psi(\tau)) \frac{d\tau}{z'(\psi(\tau))} = \frac{\varphi(\psi(t))}{z'(\psi(t))} = u(t), \quad t \in \mathbb{R}. \quad (8)$$

С помощью соотношений для замены переменных в  $\delta$ -функции, например

$$\int_{-\infty}^{\infty} a(s)\delta(s^2 - c^2 - t)ds = \int_{-\infty}^{\infty} a(s)[\delta(s - \sqrt{c^2 + t}) + \delta(s + \sqrt{c^2 + t})] \frac{1}{2\sqrt{c^2 + t}} dt =$$

$$= \frac{1}{2\sqrt{c^2 + t}} [a(\sqrt{c^2 + t}) + a(-\sqrt{c^2 + t})],$$

получены тестовые примеры, для которых известны правая часть и решение.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Методы решения некорректных задач* / [П.П.Забрейко, А.И.Кошелев, М.И.Красносельский и др.]. – М.: Наука, 1968. – 448с.
- [2] Тихонов А.Н. *О решении некорректно поставленных задач и методе регуляризации* / А.Н.Тихонов. – ДАН СССР, 1963. – С.501-504.
- [3] *Регулярирующие алгоритмы и априорная информация* / [А.Н.Тихонов, А.В.Гончарский, В.В.Степанов, А.Г.Ягола]. – М.:Наука, 1983. – 200с.
- [4] Гельфанд И.М. *Обобщенные функции и действия над ними* / И.М. Гельфанд, Т.Е. Шиллов. – М.:ФМ, 1959. – 470с.

## Спектральные свойства колинейных операторов

*Шацкая И. С.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
 ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
 КАФЕДРА АЛГЕБРЫ И ФУНКЦИОНАЛЬНОГО АНАЛИЗА (ГРУППА 500)

#### ВВЕДЕНИЕ

Пусть  $\mathcal{H}$  — кватернионный гильбертов модуль. Теорема, известная у физиков как теорема Вигнера, а у математиков как бесконечномерный аналог фундаментальной теоремы проективной геометрии утверждает, что каждая симметрия в  $\mathcal{H}$  осуществляется через некоторый коунитарный оператор  $U$ . И обратно, каждый коунитарный оператор порождает некоторую симметрию в  $\mathcal{H}$ . Эти результаты говорят о важности исследования классов колинейных операторов в кватернионных гильбертовых модулях.

#### 1. КОЛИНЕЙНЫЕ ОПЕРАТОРЫ

##### 1.1. Определение и некоторые свойства колинейных операторов.

Пусть  $\mathbb{H}$  — тело кватернионов. Как известно,  $\mathbb{H}$ -модули отличаются разнообразием классов линейных операторов. Так, аддитивный оператор  $A$ , действующий в правом кватернионном модуле  $\mathcal{H}$ , называется *линейным*, если  $A(xq) = (Ax)q$  для любого вектора  $x \in \mathcal{H}$  и для любого кватерниона  $q \in \mathbb{H}$ .

Если в этом определении заменить тело  $\mathbb{H}$  на некоторое его  $\mathbb{R}$ -содержащее подполе  $\mathbb{F}$ , то мы получим так называемый класс  $\mathbb{F}$ -линейных операторов.

**Определение 1.** *Аддитивный оператор, действующий в кватернионном модуле  $\mathcal{H}$ , называется **колинейным**, если существует автоморфизм  $\alpha$  алгебры кватернионов  $\mathbb{H}$  такой, что  $A(xq) = (Ax)\alpha(q)$  для любых  $x \in \mathcal{H}$ ,  $q \in \mathbb{H}$ .*

Заметим, что всякий автоморфизм в  $\mathbb{H}$  является внутренним, поэтому  $\alpha(q) = \bar{w}qw$ ,  $|w| = 1$ ,  $w \in \mathbb{F}$ . Такой автоморфизм будем обозначать  $\alpha_w$ .

Найдем связь между классом колинейных операторов и описанными выше классами линейных операторов.

**Предложение 1.** *Если  $A$  — колинейный оператор с автоморфизмом  $\alpha_w$ , то  $A$  —  $\mathbb{F}$ -линейный оператор, где поле  $\mathbb{F}$  есть  $\mathbb{R}$ -содержащее подполе, порождённое кватернионом  $w$ .*

*Доказательство.* Пусть  $\mathbb{F} = \mathbb{R}\langle 1, w \rangle$ ,  $q \in \mathbb{F}$ . Тогда  $\alpha_w(q) = \bar{w}qw = q$ , и, следовательно,  $A(xq) = (Ax)q$  для любого кватерниона  $q \in \mathbb{F}$ .  $\square$

**Предложение 2.** *Всякий колинейный оператор является кратным некоторого линейного оператора.*

*Доказательство.* Пусть  $A$  — колинейный оператор с автоморфизмом  $\alpha_w$ . Тогда оператор  $B = R_{\bar{w}}A$  является линейным. Действительно,

$$\begin{aligned} B(xq) &= (R_{\bar{w}}A)(xq) = A(xq)\bar{w} = (Ax)\bar{w}qw\bar{w} = (Ax)\bar{w}q = (A(x\bar{w}))q = \\ &= (R_{\bar{w}}Ax)q = (Bx)q. \end{aligned}$$

При этом  $A = R_w B$ .  $\square$

Пусть теперь  $\mathcal{H}$  — кватернионный банахов модуль. Тогда множество всех линейных ограниченных операторов  $L(\mathcal{H})$  является вещественной алгеброй. Множество всех  $\mathbb{F}$ -линейных ограниченных операторов  $L_{\mathbb{F}}(\mathcal{H})$  составляет алгебру над полем  $\mathbb{F}$ . Обозначим через  $L_{\omega}(\mathcal{H})$  множество ограниченных операторов, колинейных относительно автоморфизма  $\alpha_w$ .

**Предложение 3.** *Множество  $L_{\omega}(\mathcal{H})$  является  $\mathbb{R}$ -подмодулем в алгебре  $L_{\mathbb{F}}(\mathcal{H})$ .*

*Доказательство.* Действительно, для любых  $A, B \in L_{\omega}(\mathcal{H})$

$$(A + B)(xq) = A(xq) + B(xq) = (Ax)\alpha(q) + (Bx)\alpha(q) = ((A + B)x)\alpha(q).$$

Следовательно,  $A + B \in L_{\omega}(\mathcal{H})$ .



Согласно определению операции умножения оператора, действующего в кватернионном модуле, на скаляр  $p$  из  $\mathbb{H}$ ,

$$(pA)x := (Ax)p.$$

В этом случае оператор  $pA$  будет принадлежать алгебре  $L_\omega(\mathcal{H})$  только в том случае, когда  $p$  — вещественное. В самом деле,  $(pA)(xq) = ((pA)x)\alpha(q)$  тогда и только тогда, когда  $\alpha(q)p = p\alpha(q)$ . Последнее же равенство будет иметь место для любого кватерниона  $q$  только в том случае, когда  $p$  — вещественное.

Очевидно, произведение двух операторов из  $L_\omega(\mathcal{H})$  уже не является линейным оператором относительно автоморфизма  $\alpha_w$ . Следовательно,  $L_\omega(H)$  является  $\mathbb{R}$ -подмодулем в алгебре  $L_{\mathbb{F}}(H)$ .  $\square$

**1.2. Спектр линейного оператора.** Пусть  $A$  — линейный (соответственно  $\mathbb{F}$ -линейный) ограниченный оператор, действующий в  $\mathbb{H}$ -модуле  $\mathcal{H}$ ,  $R_q x := xq$ .

Число  $q \in \mathbb{H}$  назовем **регулярной точкой** оператора  $A$ , если вещественно линейный оператор  $A - R_q$  непрерывно обратим в алгебре вещественно линейных операторов. Множество регулярных точек оператора  $A$  назовем **резольвентным множеством** и обозначим  $\rho(A)$ . Множество  $\sigma(A) = \mathbb{H} \setminus \rho(A)$  назовем **спектром** оператора  $A$ .

В алгебре линейных (соответственно  $\mathbb{F}$ -линейных) ограниченных операторов можно ввести следующую классификацию спектра. Кватернион  $q$  принадлежит

- (1) *точечному спектру* оператора  $A$ , если  $\ker(A - R_q) \neq \{0\}$ ;
- (2) *остаточному спектру* оператора  $A$ , если  $\ker(A - R_q) = \{0\}$ ,  $\overline{R(A - R_q)} \neq \mathcal{H}$ ;
- (3) *непрерывному спектру* оператора  $A$ , если  $\ker(A - R_q) = \{0\}$ ,  $\overline{R(A - R_q)} = \mathcal{H}$ , но оператор  $(A - R_q)^{-1}$  неограничен в  $\mathcal{H}$ .

Точечный, непрерывный и остаточный спектры оператора  $A$  будем обозначать соответственно  $\sigma_p(A)$ ,  $\sigma_r(A)$ ,  $\sigma_c(A)$ .

Напомним, что множество  $K(q) = \{uq\bar{u} \mid u \in \mathbb{H}, |u| = 1\}$  есть класс сопряженности элемента  $q$  в мультипликативной группе  $\mathbb{H}^*$ . Известно, что всякий кватернион  $q$  принадлежит множеству регулярных точек (точечному, непрерывному, остаточному спектрам соответственно) вместе со всем своим классом сопряженности.

Пусть  $A \in L_\omega(\mathcal{H})$ . Тогда согласно предложению 2 имеет место равенство  $A = R_w B$ , где  $B$  — линейный оператор, откуда

$$A - R_q = R_w B - R_q = R_w(B - R_{\bar{w}} R_q) = R_w(B - R_{q\bar{w}}).$$

Поэтому справедливо утверждение.

**Предложение 4.** Пусть  $\mathcal{H}$  — банахов  $\mathbb{H}$ -модуль,  $A \in L_\omega(\mathcal{H})$ ,  $A = R_\omega B$ . Тогда

- (i)  $q \in \rho(A)$  тогда и только тогда, когда  $q\bar{w} \in \rho(B)$ ;
- (ii)  $q \in \sigma_t(A)$  тогда и только тогда, когда  $q\bar{w} \in \sigma_t(B)$ ,  $t \in p, c, r$ .

Будем говорить, что ненулевой кватернион  $p$  косопряжён кватерниону  $q$ , если существует кватернион  $u \in \mathbb{H}$ ;  $|u| = 1$  такой, что  $p = uq\alpha(\bar{u})$ .

Это отношение косопряжённости есть отношение эквивалентности, которое разбивает все множество  $\mathbb{H}$  на классы эквивалентности

$$C(q) = \{uq\alpha(\bar{u}) | u \in \mathbb{H}, |u| = 1\}.$$

**Предложение 5.** Пусть  $\mathcal{H}$  — банахов  $\mathbb{H}$  - модуль,  $A \in L_\omega(\mathcal{H})$ ,  $A = R_\omega B$ . Тогда справедливо следующее:

- (a) если  $q \in \rho(A)$ , то  $C(q) \subset \rho(A)$ ;
- (b) если  $q \in \sigma_t(A)$ , то  $C(q) \subset \sigma_t(A)$ ,  $t \in p, c, r$ .

*Доказательство.* Пусть  $q \in \rho(A)$ . Тогда на основании предложения 4  $q\bar{w} \in \rho(B)$ . Следовательно, для любого кватерниона  $u$ ,  $|u| = 1$ ,  $uq\bar{w}\bar{u} \in \rho(B)$ , откуда  $uq\bar{w}\bar{u}w = uq\alpha(\bar{u}) \in \rho(A)$ .

Утверждение (b) доказывается аналогично.  $\square$

Множество всех собственных векторов коллинейного оператора  $A$ , соответствующих собственному значению  $q$ , обозначим  $E_q(A)$ .

**Предложение 6.** Пусть  $\mathcal{H}$  — банахов  $\mathbb{H}$  - модуль,  $A \in L_\omega(\mathcal{H})$ ,  $A = R_\omega B$ . Если  $q \in \sigma_p(A)$ , то  $E_q(A) = E_{q\bar{w}}(B)$ .

*Доказательство.* Пусть  $Ax = xq$ ,  $x \neq 0$ . Действуя на это равенство оператором  $R_{\bar{w}}$ , получим  $Bx = x(q\bar{w})$ . Следовательно,  $x \in E_{q\bar{w}}(B)$ .  $\square$

**Следствие 1.** Множество  $E_q(A)$  есть  $\mathbb{F}$ -линейный модуль, где  $\mathbb{F}$  — поле, порождённое кватернионом  $q\bar{w}$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] S.L.Adler, G.G.Emch *A rejoinder on quaternionic projective representations*, arXiv:hep-th/9704121v3 18 Apr 1997, 1–5
- [2] Карпенко И.И., Тышкевич Д.Л. *Спектральные свойства линейных операторов над гильбертовыми кватернионными бимодулями*, Математичні СТУДІЇ, Том 30, No.1, 2008, 67–83.

## Секция ПРИКЛАДНОЙ МАТЕМАТИКИ

### Применение технологии Swing для разработки кросс-платформенных приложений клиент-сервер

*Абоимов С. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Рассмотрим задачу программной реализации графического, кросс-платформенного приложения клиент-сервер, позволяющего осуществлять обмен сообщениями и файлами. Сервер должен работать как своеобразный "почтовый сервер" который хранит все сообщения и файлы. Клиентская часть функционирует на половину как почтовая программа, на половину как чат. То есть, клиент позволяет отсылать сообщения и файлы пользователям офф-лайн, а также быстро обмениваться текстовыми сообщениями с пользователями, находящимися он-лайн.

Для реализации приложения из всех известных платформ разработки была выбрана платформа JDK, обеспечивающая переносимость проектов, независимость от используемой ОС, а также имеющая достаточно развитую на сегодняшний день библиотеку компонентов. Для графического интерфейса был выбран Swing, поскольку аналог SWT от IBM плохо себя проявил при переносе на другие платформы. Это выразилось в отсутствии общих библиотек, которые способны работать сразу со всеми платформами (усложняется процесс разработки, если пользователи пишут на разных платформах), отсутствие мультимониторов (графическое приложение обязано постоянно синхронизировать потоки с главным дисплеем), слабо функциональные компоненты, что повлекло за собой отсутствие хорошего визуального редактора, который ускоряет процесс разработки, и наконец полное не соответствие шрифтов. В качестве сервера базы данных был выбран MySQL ввиду простоты его использования (в отличие от Oracle и Db2) кроссплатформенности (в отличие от MSSQL), наличия хорошей документальной поддержки и развитой библиотеки компонентов.

Теперь подробнее остановимся на архитектуре библиотеки Swing.

Одной из основных особенностей архитектуры Swing является поддержка модели MVC: Model-View-Controller. Если необходимо создать таблицу, вы описываете контейнер данных, реализующий интерфейс доступа TableModel

для кода отрисовки таблицы. В зависимости от задачи, модель данных может быть создана специально для таблицы (например можно использовать стандартный `DefaultTableModel`), а можно иметь сложное собственное хранилище и присоединить к нему реализацию интерфейса доступа. Во втором случае часто удобнее сделать класс "посредник обеспечивающий доступ к данным для кода отрисовки таблиц, нежели делать реализацию непосредственно в контейнер данных. Теперь, если необходимо нарисовать таблицу, мы создаем новый объект "редактор"`JTable`, передавая ему объект, реализующий интерфейс `TableModel`.

Модель данных, непосредственно или через класс-посредник, общается двухсторонним образом с редактором. Редактор запрашивает у модели параметры таблицы и сами данные, а модель в свою очередь уведомляет редактор о происходящих в ней изменениях. Для этого редактор (`JTable`) регистрируется на получение событий об изменении модели. К одной модели данных может быть создано сколько угодно редакторов. Изменения, сделанные в одном редакторе, попадут в модель. Модель, в свою очередь разошлет события `TableModelEvent` всем редакторам (через интерфейс `TableModelListener`). Если у модели есть еще какие-то другие редакторы, то она расскажет им на их собственном "языке" о произошедших внутри нее изменениях.

По аналогии сделаны практически все компоненты Swing'a, включая всевозможные контейнеры компонентов. Однако в этой архитектуре имеется одно тонкое место, о котором разработчики Swing не подумали. Имея на экране какое-либо сложное окно GUI, мы внутри имеем десятки или сотни компонентов, связанных между собой механизмом рассылки событий. У каждого компонента есть список объектов, ожидающих получения событий о его изменении (на самом деле очередей обычно несколько, для каждого типа события создается своя очередь, но это сейчас не важно).

В случае, если мы хотим иметь долгоживущее хранилище данных и периодически создавать новые редакторы для этой модели, каждый новый редактор будет регистрироваться на получение событий изменения модели. Если ничего специально не предпринимать, то Listener'ы событий будут накапливаться в списке ожидающих событий модели. За каждым Listener'ом будет стоять целая цепочка компонентов, отвечающих за отрисовку данных на экране, не важно что компоненты давно уже не показываются на экране и никому не нужны. С точки зрения компонента Swing нельзя определить момент, когда компонент более уже никому не нужен и должен быть удален - какого-либо аналога деструктора объекта у компонентов Swing нет (такого,

как например метод `close()` у объекта `File` или `SqlConnection`). Нельзя понять, то ли мы временно спрятали объект, то ли выкинули его навсегда.

Таким образом, в полученной схеме мы будем по мере жизни программы терять память, а вместе с этим и процессорное время, которое будет тратиться на рассылку событий никому ненужным компонентам. Наиболее разумным решением в этой ситуации, по-видимому, является добавление явного "деструктора" в какой-нибудь из корневых объектов иерархии Swing, но это скорее всего не будет реализовано разработчиками. Создавать "деструкторы" к компонентам Swing наследованием не слишком удобно и полностью не возможно (т.к. Swing внутри имеет весьма сложную архитектуру, и часть этой самой архитектуры скрыта механизмом инкапсуляции).

Можно следить за событиями удаления компонента с экрана и при этом удалять свои `Listener`'ы из модели, а при восстановлении - обратно восстанавливать. Это, в принципе, решает проблему, но является, по большому счету, "костылем". К тому же это приводит к дополнительным нагрузкам на процессор при возвращении объекта обратно на экран, что иногда неприемлемо.

Разработчики Java 6 предлагают использовать "висячие ссылки" `WeakReference` для связи объекта `Listener`'а с объектом, ожидающим получения события. Таким образом, неиспользуемый никому ненужный редактор может быть удален сборщиком мусора, и вместе с этим при приходе следующего события `Listener` сможет разрегистриваться у модели данных.

Описанные выше особенности библиотеки Swing позволили разработать эффективное кроссплатформенное приложение `qMessenger`, реализующее все необходимые функции по поддержке документооборота и информационного обмена.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] <http://java.sun.com/products/jfc/tsc/articles/architecture/>
- [2] <http://java.sun.com/j2se/1.5.0/docs/>
- [3] <http://www.eclipse.org/swt/docs.php>

## Разработка объектных графических редакторов

*Вьшарев И. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Работа посвящена построению набора классов, компонент и алгоритмов для реализации приложения с графическим пользовательским интерфейсом,

а также решению архитектурных проблем, возникающих при проектировании. В качестве примера рассматривается приложение „Optics“, разрабатываемое в рамках дипломной работы.

При разработке приложений не всегда хватает стандартных средств. Основная идея программы „Optics“ заключается в том, чтобы пользователь мог создавать и легко модифицировать схемы оптических приборов. Реализовать это довольно трудно, если все элементы рисовать графическими примитивами. Необходимой гибкости можно добиться, если отдельные элементы будут представлять собой графические компоненты (такие как панели, кнопки и прочее). Тогда их можно будет легко добавлять, удалять, перемещать и изменять размеры. Здесь и приходится расширять существующие или создавать свои.

В приложении „Optics“ у всех оптических инструментов (выпуклая и вогнутая линзы, зеркало, непрозрачная поверхность, источник и луч света) отделены друг от друга описывающие их данные, логика и представление. Классы, которые объединяют в себе эти составляющие, находятся в иерархии, которая сходится к классу, реализующему панель. Это наделяет созданный компонент всеми возможностями графического элемента. Т.к. каждый оптический инструмент выглядит по-своему, то панель, которая будет являться этим компонентом, должна быть прозрачной. Как себя рисовать компонент узнает у своего представления, данные, которые нужны для его отображения, находятся в модели. Иерархия классов позволяет избежать повторений в описании модели и реализации логики.

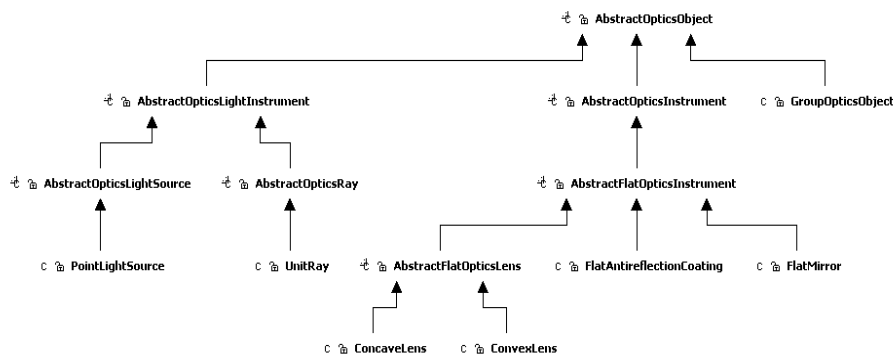


Рис. 1. UML-диаграмма иерархии классов моделей оптических инструментов.

Например, линзы и зеркала – плоские объекты (`AbstractFlatOpticsInstrument` на рис. 1), а значит, у них есть общие свойства (длина, координаты крайних точек). У линз есть дополнительные свойства (оптический центр линзы, фокусы), следовательно, модель для них нужно расширить. Чем лучше составлена иерархия, тем проще расширять приложение. Если нужно добавить новый инструмент в приложении, достаточно создать необходимые классы, найти их место в иерархии.

И так, есть возможность создания, добавления и удаления компонент произвольного вида. А как ими управлять, изменять параметры и размеры? Всегда можно дать пользователю возможность ввода с клавиатуры, но набирать каждое значение по отдельности неудобно. Лучше предоставить альтернативный способ управления мышью. Реализовать эту функциональность можно, используя маркеры управления. Это небольшие вспомогательные компоненты, которые привязаны к другому, и при перемещении влияют на его свойства. Удобно инкапсулировать возможность перетаскивания маркеров управления. Реализовать их влияние на свойства других компонент можно, оставив нереализованный (абстрактный) метод, описывающий логику. Маркеры управления не могут существовать сами по себе, т.к. они должны управлять каким-нибудь компонентом. Так же они могут выполнять различные функции - перемещать, изменять размер, вращать.

Хотя ввод значений с клавиатуры не так удобен, но это дает пользователю возможность задавать данные более точно. Одно из решений этой задачи является использование инспектора объектов. Это элемент графического интерфейса, отображающий и позволяющий редактировать свойства выбранного объекта. Так же инспектор объектов обладает функциональностью навигации по объектам, где пользователь может из списка выбрать нужный. Инспектор объектов должен следить за изменением модели компонент. Чтобы изменять данные, можно использовать механизмы рефлексии. Если у пользователя есть возможность выделять компоненты мышью, то инспектор объектов должен показывать свойства выделенных.

Основным элементом графического редактора, является рабочий холст. С точки зрения пользователя, это пространство на котором и производится рисование или создание чертежей. С точки зрения разработчика, рабочий холст - это контейнер, который может в себе содержать другие компоненты. Если компоненты расположить таким образом, что они будут перекрывать друг друга, то возникает проблема доступа к перекрытому объекту. Для решения

этой проблемы используется многослойный рабочий холст. Нижний слой содержит добавленные объекты. А его фон предназначен для рисования изображений, не связанных с объектами (такими как координатная сетка, траектория луча). Чтобы избежать лишних пересчетов и построений, фон нижнего слоя хранится в буфере. Когда нужно изменить изображение, буфер обновляется. Верхний слой - прозрачный, содержит только маркеры управления всех добавленных объектов. Это и обеспечивает доступ ко всем компонентам независимо от их расположения.



Рис. 2. Многослойный рабочий холст.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Портянкин И. *Swing. Эффективные пользовательские интерфейсы.*
- [2] Вязовик Н.А. *Программирование на Java.*
- [3] Шилдт Г. *Swing. Руководство для начинающих.*

## Моделирование гладких параметрических кривых и поверхностей с использованием интерполяции по Эрмиту

*Клевцова В. Н.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Обобщенным интерполированием по Эрмиту в численных методах называется такой вариант интерполирования, когда в узлах заданы значения функции и некоторое количество ее производных.

Целью работы является использование этого подхода в некоторых частных случаях для построения параметрических кривых и поверхностей. Результаты исследования могут быть применены:



- при моделировании поверхности физических объектов;
- для решения задач плоского и пространственного интерполирования;

Рассматривается задача, когда нужно найти функцию  $r(u)$ ,  $0 \leq u \leq 1$ , которая удовлетворяет 5 условиям:  $r(0)$ ,  $r(1)$ ,  $r(u^*)$ ,  $\dot{r}(0)$ ,  $\dot{r}(1)$ . Тогда необходимо формировать  $r(u) = \sum_{i=0}^4 a_i u^i$ . Выбор многочлена в качестве интерполирующей функции логичен, т. к. данный подход вытекает из классических задач интерполирования.

В  $r(u)$ , коэффициенты  $a_i$  зависят от значения радиус-вектора  $r$  и его производной по  $u - \dot{r}$  на обоих концах сегмента  $r(0)$ ,  $r(1)$ ,  $\dot{r}(0)$ ,  $\dot{r}(1)$  и от радиус-вектора на сегменте  $r(u^*)$ , где  $0 < u^* < 1$  [1]

Приведем алгоритм нахождения коэффициентов  $\bar{a}_i$  для данных условий.

$$\begin{cases} r(0) = a_0; \\ r(1) = a_0 + a_1 + a_2 + a_3 + a_4; \\ r(x) = a_0 + ua_1 + u^2a_2 + u^3a_3 + u^4a_4; \\ \dot{r}(0) = a_1; \\ \dot{r}(1) = a_1 + 2a_2 + 3a_3 + 4a_4; \end{cases}$$

Найдем неизвестные  $a_0, a_1, a_2, a_3, a_4$  :

$$a_0 = r(0);$$

$$a_1 = \dot{r}(0);$$

$$a_2 = \frac{r(0)(-3x^4+4x^3-1)+r(1)(3x^4-4x^3)+r(x)+\dot{r}(0)(-2x^4+3x^3-x)+\dot{r}(1)(-2x^4+3x^3-x)}{x^2(1-x)^2};$$

$$a_3 = \frac{r(0)(2x^4-4x^2+2)+r(1)(-2x^4+4x^2)-2r(x)+\dot{r}(0)(x^4-3x^2+2x)+\dot{r}(1)(x^4-x^2)}{x^2(1-x)^2};$$

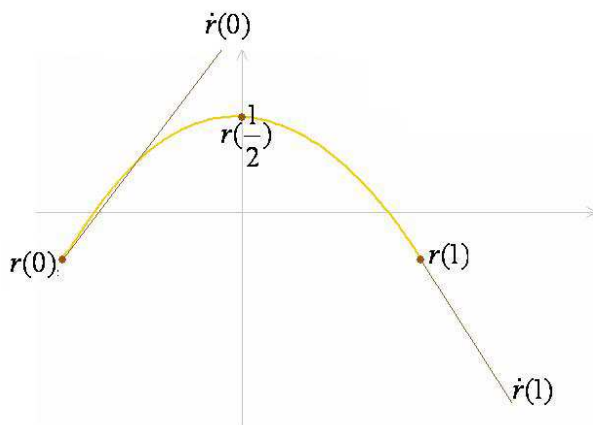
$$a_4 = \frac{r(0)(-2x^3+3x^2-1)+r(1)(2x^3-3x^2)+r(x)+\dot{r}(0)(-x^3+2x^2-x)+\dot{r}(1)(-x^3+x^2)}{x^2(1-x)^2};$$

Следовательно, многочлен  $r(u)$  примет вид:

$$\begin{aligned} r(u) = & r(0) \left( 1 + \frac{-3x^4+4x^3-1}{x^2(1-x)^2} u^2 + \frac{2x^4-4x^2+2}{x^2(1-x)^2} u^3 + \frac{-2x^3+3x^2-1}{x^2(1-x)^2} u^4 \right) + \\ & + r(1) \left( \frac{3x^4-4x^3}{x^2(1-x)^2} u^2 + \frac{-2x^4+4x^2}{x^2(1-x)^2} u^3 + \frac{2x^3-3x^2}{x^2(1-x)^2} u^4 \right) + \\ & + r(x) \left( \frac{1}{x^2(1-x)^2} u^2 + \frac{-2}{x^2(1-x)^2} u^3 + \frac{1}{x^2(1-x)^2} u^4 \right) + \end{aligned}$$

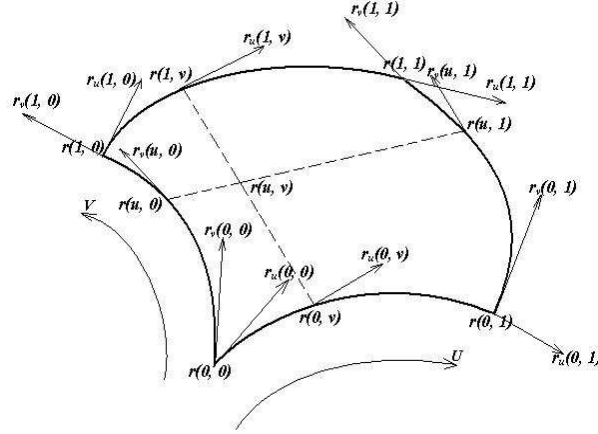
$$\begin{aligned}
& +\dot{r}(0) \left( u + \frac{-2x^4+3x^3-x}{x^2(1-x)^2}u^2 + \frac{x^4-3x^2+2x}{x^2(1-x)^2}u^3 + \frac{x^3+2x^2-x}{x^2(1-x)^2}u^4 \right) + \\
& +\dot{r}(1) \left( \frac{-x^4+x^3}{x^2(1-x)^2}u^2 + \frac{x^4-x^2}{x^2(1-x)^2}u^3 + \frac{-x^3+x^2}{x^2(1-x)^2}u^4 \right)
\end{aligned}$$

Рассмотрим в рамках постановки задачи рисунок при  $u^* = \frac{1}{2}$



Основной целью задачи является построение соответствующего трехмерного объекта. Тогда  $r(u, v) = (1, u, u^2, u^3) \cdot F \cdot R \cdot F^T \cdot \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix}$ ,  $0 \leq u, v \leq 1$ , где теперь в отличие от классического подхода Фергюсона [2] матрицы  $F$  и  $R$  пятого порядка.

Модель предлагаемого участка поверхности изображена на рисунке.

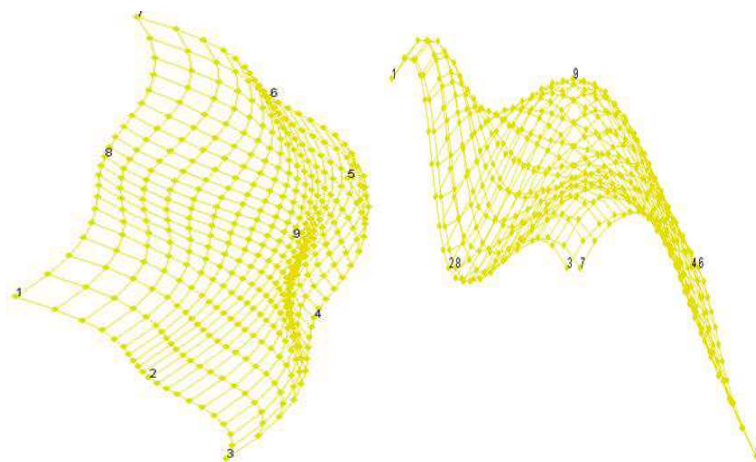


$$F(u^*) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{-3u^{*4}+4u^{*3}-1}{u^{*2}(1-u^*)^2} & \frac{3u^{*4}-4u^{*3}}{u^{*2}(1-u^*)^2} & \frac{1}{u^{*2}(1-u^*)^2} & \frac{-2u^{*4}+3u^{*3}-u^*}{u^{*2}(1-u^*)^2} & \frac{-u^{*4}+u^{*3}}{u^{*2}(1-u^*)^2} \\ \frac{2u^{*4}-4u^{*2}+2}{u^{*2}(1-u^*)^2} & \frac{-2u^{*4}+4u^{*2}}{u^{*2}(1-u^*)^2} & \frac{-2}{u^{*2}(1-u^*)^2} & \frac{u^{*4}-3u^{*2}+2u^*}{u^{*2}(1-u^*)^2} & \frac{u^{*4}-u^{*2}}{u^{*2}(1-u^*)^2} \\ \frac{-2u^{*3}+3u^{*2}-1}{u^{*2}(1-u^*)^2} & \frac{2u^{*3}-4u^{*2}}{u^{*2}(1-u^*)^2} & \frac{1}{u^{*2}(1-u^*)^2} & \frac{-u^{*3}+2u^{*2}-u^*}{u^{*2}(1-u^*)^2} & \frac{-u^{*3}+u^{*2}}{u^{*2}(1-u^*)^2} \end{bmatrix}$$

$$R(u^*, v^*) = \begin{bmatrix} r(0, 0) & r(0, 1) & r(0, v^*) & r_v(0, 0) & r_v(0, 1) \\ r(1, 0) & r(1, 1) & r(1, v^*) & r_v(1, 0) & r_v(1, 1) \\ r(u^*, 0) & r(u^*, 1) & r(u^*, v^*) & r_v(u^*, 0) & r_v(u^*, 1) \\ r_u(0, 0) & r_u(0, 1) & r_u(0, v^*) & r_{uv}(0, 0) & r_{uv}(0, 1) \\ r_u(1, 0) & r_u(1, 1) & r_u(1, v^*) & r_{uv}(1, 0) & r_{uv}(1, 1) \end{bmatrix}$$

где  $0 < u^*, v^* < 1$

Тогда использование значений  $u^* = \frac{1}{2}, v^* = \frac{1}{2}$  позволяет посторить следующий четырехугольный участок поверхности:



Т.к. значения  $u^*$ ,  $v^*$  можно выбирать любыми на  $(0,1)$ , то точка, определяемая парой  $(u^*, v^*)$ , может быть выбрана практически в любом месте четырехугольной порции поверхности. Поэтому рассмотренную модель можно использовать в тех случаях, когда нужно локально модифицировать фрагмент поверхности так, чтобы он обязательно прошел через некую новую точку, появление которой обусловлено требованиями проектировщиков. Кроме того, можно использовать указанную методику в классических задачах трехмерного интерполирования таблично заданной функции.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Ковтун О.М. *Поліноміальні сплайни четвертого ступеня*. „Прикладна геометрія та інженерна графіка“ вип. 79, 2004, стр. 239 - 243.
- [2] Фокс А., Пратт М. „*Вычислительная геометрия*“. Пер. с англ. - М.: Мир, 1982 - 304 с.

## Защита информации в компьютерных сетях

*Кошценко А. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Появление новых информационных технологий и развитие мощных компьютерных систем хранения и обработки информации повысили уровни защиты информации и вызвали необходимость в том, чтобы эффективность защиты информации росла вместе со сложностью архитектуры хранения данных.

Таким образом, угроза защиты информации сделала средства обеспечения информационной безопасности одной из обязательных характеристик информационной системы.

В современном мире наиболее актуальными сферами применения криптографической защиты информации - это технологии, связанные с электронной цифровой подписью, а также обеспечение безопасности персональных данных.

Большинство средств защиты информации базируется на использовании криптографических шифров и процедур шифрования - дешифрования.

Методы шифрования в зависимости от структуры применяемых ключей делятся на [1]:

- симметричное шифрование: противнику может быть известен алгоритм шифрования, но неизвестен ключ. В симметричном шифровании ключ одинаковый для отправителя и для получателя сообщения.

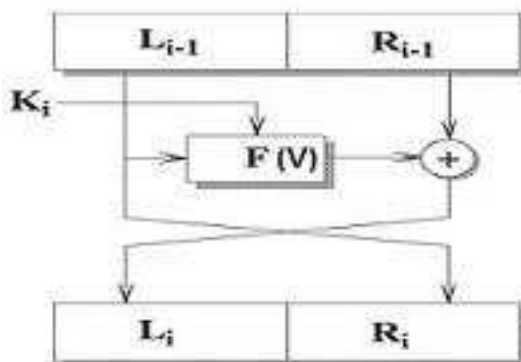
- асимметричное шифрование: противнику может быть известен алгоритм шифрования, и, возможно, открытый ключ, но неизвестен закрытый ключ, известный только получателю.

Алгоритмы симметричного шифрования различаются способом, которым обрабатывается исходный текст. Возможно шифрование блоками или шифрование потоком.

Блочный алгоритм преобразовывает  $n$ -битный блок незашифрованного текста в  $n$ -битный блок зашифрованного текста. Число блоков длины  $n$  равно  $2n$ . Для того чтобы преобразование было обратимым, каждый из таких блоков должен преобразовываться в свой уникальный блок зашифрованного текста. При незначительной длине блока такая подстановка плохо скрывает статистические особенности незашифрованного текста. Если блок имеет длину 64 бита, то он уже хорошо скрывает статистические особенности исходного текста. Но в данном случае преобразование текста не может быть произвольным в силу того, что ключом будет являться само преобразование, что исключает эффективную как программную, так и аппаратную реализации.

Наиболее широкое распространение получили сети Фейстеля, так как, с одной стороны, они удовлетворяют всем требованиям к алгоритмам симметричного шифрования, а с другой стороны, достаточно просты и компактны.

Сеть имеет следующую структуру. Входной блок делится на несколько равной длины подблоков, называемых ветвями. В случае, если блок имеет длину 64 бита, используются две ветви по 32 бита каждая. Каждая ветвь обрабатывается независимо от другой, после чего осуществляется циклический сдвиг всех ветвей влево. Такое преобразование выполняется за несколько циклов



или раундов. В случае двух ветвей каждый раунд имеет структуру, показанную на рисунке:

Функция  $F$  называется образующей. Величины  $V$  именуется параметрами сети. Каждый раунд состоит из вычисления функции  $F$  для одной ветви и побитового выполнения операции XOR результата  $F$  с другой ветвью. После этого ветви меняются местами. Считается, что оптимальное число раундов - от 8 до 32. Важно то, что увеличение количества раундов значительно увеличивает криптостойкость алгоритма. Возможно, эта особенность и повлияла на столь активное распространение сети Фейстеля, так как для большей криптостойкости достаточно просто увеличить количество раундов, не изменяя сам алгоритм. В последнее время количество раундов не фиксируется, а лишь указываются допустимые пределы [2].

Сеть Фейстеля является обратимой даже в том случае, если функция  $F$  не является таковой, так как для дешифрования не требуется вычислять  $F^{-1}$ . Для дешифрования используется тот же алгоритм, но на вход подается зашифрованный текст, и ключи используются в обратном порядке.

В настоящее время все чаще используются различные разновидности сети Фейстеля для 128-битного блока с четырьмя ветвями. Увеличение количества ветвей, а не размерности каждой ветви связано с тем, что наиболее популярными до сих пор остаются процессоры с 32-разрядными словами, следовательно, оперировать 32-разрядными словами эффективнее, чем с 64-разрядными.

Основной характеристикой алгоритма, построенного на основе сети Фейстеля, является функция  $F$ . Различные варианты касаются также начального и конечного преобразований.

Сеть Фейстеля надежно зарекомендовала себя как криптостойкая схема произведения преобразований, и ее можно найти практически в любом современном блочном шифре. Незначительные модификации касаются обычно дополнительных начальных и конечных преобразований над шифруемым блоком. Подобные преобразования, выполняемые обычно также либо "исключающим ИЛИ" или сложением имеют целью повысить начальную рандомизацию входного текста. Таким образом, криптостойкость блочного шифра, использующего сеть Фейстеля, определяется на 95 процентов функцией  $F$  и правилом вычисления  $V$  из ключа. Эти функции и являются объектом все новых и новых исследований специалистов в области криптографии.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Блэк У. *Интернет: протоколы безопасности. Учебный курс.* – СПб.: Питер, 2001. - 288с.:ил.
- [2] Лапонина О.Р. *Основы сетевой безопасности: криптографические алгоритмы и протоколы взаимодействия.* – Интернет-университет информационных технологий - ИНТУИТ.ру, 2005. - 608 с.

## Оптимизация архитектура и проектирование программных приложений в условиях жестких ограничений ресурсов

*Левинцов Н. Р.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Разработка ПО - непростой процесс, который может включать множество компонентов. За последние 25 лет ученые выделили одиннадцать, среди которых 3 компонента напрямую связаны с разработкой архитектуры и проектированием: создание плана конструирования, разработка архитектуры ПО, детальное проектирование, выработка требований. И один компонент соответствует оптимизации, это - корректирующее сопровождение. Следовательно, проблема связана с более чем третьей частью цикла разработки программного обеспечения.

Цель работы - разработка программного комплекса и улучшение его параметров с использованием приемов оптимизации. Параметры будут описаны далее.

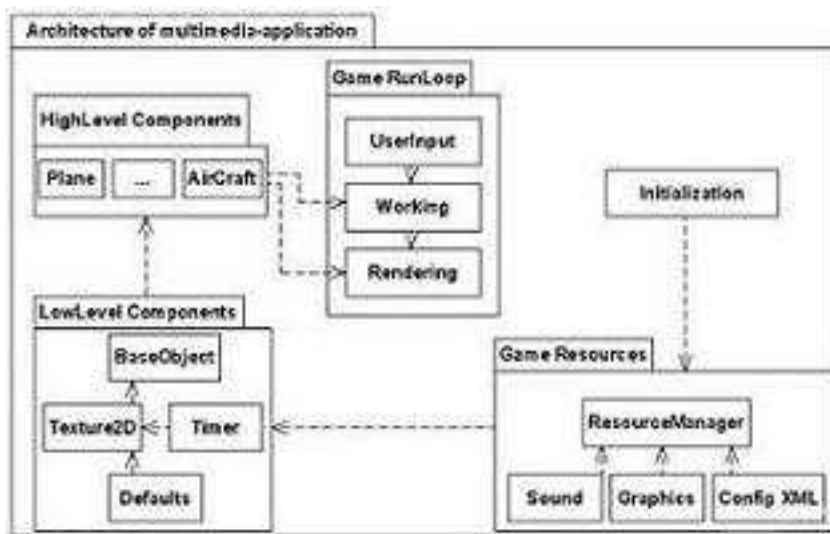


Рис. 1. Архитектура приложения.

Для реализации поставленной задачи была спроектирована модель программы, представленная на рисунке 1.

Введем следующие определения.

**Качество ПО** - это совокупность свойств, определяющих полезность программы для пользователей в соответствии с функциональным назначением и предъявленными требованиями.

**Критерий качества** - численный показатель, характеризующий степень, в которой программе присуще оцениваемое свойство.

**Характеристика качества программы** - понятие, отражающее отдельные факторы, влияющие на качество программ и поддающиеся измерению.

**Метрика качества программ** - система измерений качества программ. Эти измерения могут проводиться на уровне критериев качества программ или на уровне отдельных характеристик качества.

**Цикломатическое число.** Впервые графическое представление программ было предложено Маккейбом [1]. Основной метрикой сложности он предлагает считать цикломатическую сложность графа программы, или, как ее еще называют, цикломатическое число Маккейба, характеризующее трудоемкость тестирования программы. Для вычисления цикломатического числа Маккейба  $Z(G)$  применяется формула:



$$Z(G) = e - v + 2p,$$

где  $e$  - число дуг ориентированного графа  $G$ ;  $v$  - число вершин;  $p$  - число компонентов связности графа.

По сути  $Z(G)$  определяет число линейно независимых контуров в сильно связном графе. Иначе говоря, цикломатическое число Маккейба показывает требуемое количество проходов для покрытия всех контуров сильно связного графа или количество тестовых прогонов программы, необходимых для исчерпывающего тестирования по критерию "работает каждая ветвь". Цикломатическое число зависит только от количества предикатов, сложность которых при этом не учитывается.

Локальное быстродействие - время (в процентах от общего времени выполнения программы), затрачиваемое на реализацию какой-либо функциональности. Применяется к функциям, как основным структурным единицам программы.

Количество кадров в секунду (frame rate) - частота обновления экрана.

В результате оптимизации получены следующие улучшения по ключевым параметрам:

frame rate = 60

$Z(G)$  уменьшена на 20 процентов

средний рост локального быстродействия 30 процентов.

Работа по оптимизации продолжается.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Макконнел С. *Совершенный код*. – ISBN: 5-7502-0064-3, 2005. – 896 с.
- [2] Фаулер М. *Рефакторинг: улучшение существующего кода*. – ISBN: 5-93286-045-3, 2003. – 432 с., ил.

## Разработка интеллектуализированной системы обработки информации по составным символам

*Лемтюжникова Д. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

В работе предлагается вариант решения задачи анализа и синтеза составных символов различного происхождения с помощью интеллектуализированной системы обработки информации по символам (ИСОИС). Разработка таких ИС разбивается на ряд подзадач по информационным технологиям, методам и алгоритмам. Существенным является учёт всей имеющейся информации (археологической, исторической, научной и др.) в базе знаний ИС. Снятие неопределённости осуществляется на основе экспертных оценок, обучения и адаптации ИС к новым данным и знаниям. Эти задачи тесно связаны с задачами искусственного интеллекта [1]: распознавание образов, машинное обучение, кодирование, создание экспертных систем и баз знаний, машинный перевод, информационный поиск, машинное творчество. Опишем характерные задачи по наполнению ИСОИС на примере распознавания лигатур древних алфавитов. Задача кодирования [2] *каждого простого символа и разработки метода кодирования сложных символов при помощи набора простых алгоритмов.* Распознавание составных структур приводит к разбиению лигатур на символы заданного алфавита. Далее происходит обучение системы: распознанный символ становится частью набора базовых символов. Для этого осуществляется построение базы знаний, в которую заносятся базовые символы алфавита. Следует создать базу данных, включающую информацию о происхождении сложных символов и обстоятельств их нахождения, чтобы извлекать дополнительную информацию для уточнения символов базового алфавита и систематизации метода расшифровки на основании полученных данных. Чтобы оптимизировать процесс, ИС выполняет функции экспертной системы по использованию базы знаний и базы данных для получения достоверных заключений относительно распознавания составных символов. Задача составления осмысленного текста является самостоятельной проблемой. Для этого требуется построить языковой анализатор. Он будет включать заданный алфавит, способ словообразования, конструкции, существующие в языке и конструкции, которые существовать не могут. Можно поставить задачу машинного перевода и получить дополнительную информацию о сложных символах. Полученные данные следует поместить в базу знаний, расширяя аппарат машинного обучения. Для оптимизации данного процесса следует разработать



Рис. 1. Комбинация прямой и зеркальной Беркан (а). Комбинация Ингуза и двух Ис (б).

метод системного поиска, так как база данных и база знаний будут расширяться, что затруднит обращение к этим объектам и поиск требуемой информации. *Задача синтеза.* Пусть вышеперечисленные задачи выполнены. Тогда можно выделить некоторую закономерность построения составных символов. Необходимо учитывать тот факт, что лигатуры должны нести определённую смысловую нагрузку с учётом языковых особенностей и быть гармоничными с точки зрения графического изображения. Появляется *задача машинного творчества*, требующая особого подхода, так как природа творчества изучена меньше, чем природа интеллекта. Для программной реализации предлагается представлять символы в виде графов, закодировать их как список рёбер такого графа, ввести операции для графов, построить алгоритмы для обработки символов, составить базу данных, базу знаний и построить связи между ними, построить языковой анализатор и оптимизировать систему при помощи дополнительных алгоритмов. Особого внимания заслуживают древние алфавиты, символы которых вырезались на твёрдом основании – *руны* [3]. Из стандартного набора символов можно построить более сложный символ, называемый *вязью*, путём геометрического построения. Лигатуры, построенные при помощи прямых можно задавать матрицами, а лигатуры, построенные при помощи кривых можно задавать при помощи матриц и кривых Безье. Базовые символы могут склеиваться между собой, нанизываться друг на друга, при этом каждый из них может быть перевёрнут, зеркально отражён и изменён в размере. Выделим следующие свойства рунического алфавита: неполнота, избыточность первого ранга, избыточность второго ранга и избыточность  $n$ -ного ранга. Суммарный символ обладает свойством неполноты, если способ его разбиения не единственный. Рассмотрим следующий пример: сложный символ нужно разбить на простые символы (рис.1а,б). Если символ обладает свойством неполноты, мы получим несколько вариантов расшифровки. Избыточностью первого ранга будем называть такое свойство, когда одни простые символы алфавита содержат в себе другие простые символы из этого же алфавита. Рассмотрим следующий пример (рис.2а). Учитывая избыточность первого ранга при расшифровке сложного символа, можно избежать мнимых комбинаций. Избыточностью второго ранга будем называть

свойство, когда при сплетении нескольких рун в итоговом символе содержатся новые руны, и при этом учитывается избыточность первого ранга. Рассмотрим следующий пример (рис.2б). Уровень составного символа определяется минимальным количеством простых символов, из которых он состоит. Избыточностью  $n$ -ого ранга будем называть свойство, когда при сплетении нескольких составных символов в итоговом символе содержатся новые руны, и при этом учитывается избыточность  $(n-1)$ -ого ранга, при этом  $n$  соответствует уровню составного символа. Когда мы получаем расшифровку составного символа, мы ставим метку его уровня для того, чтобы в дальнейшем при расшифровке более сложного символа не получить такие базовые символы, комбинации которых составляют данный символ. Зная уровень символа, мы можем применять свойство избыточности  $n$ -ого ранга. Составляя из базовых символов алфавит, мы определяем время возникновения того или иного предмета, на котором начерчен символ, а также интерпретируем значение надписи. Символ, помещённый в базу знаний, должен быть представлен в виде линии, которая описывается математически как единый объект. Рассмотрим такой способ формирования математической интерпретации, как праматрица [4]. С её помощью задаётся любой алфавитный символ, как выделенный сегмент. Выделяя множество вершин и множество рёбер между вершинами (рис.3), рассмотрим праматрицу как неориентированный граф [5]. Обозначим рёбра полученного графа  $a[i][j]$  и составим для данного графа список рёбер. Таким образом мы получим матрицу размерности  $13 \times 13$ . Заполним эту матрицу по следующему принципу: 0 – нет пути из вершины  $i$  в вершину  $j$ , а 1 – если есть. Запишем результат в следующем виде: составим матрицу вертикальных линий, матрицу диагональных линий с наклоном влево и матрицу диагональных линий с наклоном вправо, учитывая, что наш граф является неориентированным ( $a[i][j] = a[j][i]$ ), а значит при кодировке мы можем использовать только  $a[i][j]$ . Мы можем задать каждый простой символ тремя матрицами. Например  $Fehu()$ ;



Рис. 2. Беркана содержит Райдо, Райдо содержит Вуньо, Вуньо содержит Лагуз, Лагуз содержит Ису.(а). Полученный символ содержит Наутиз, но при этом ни Соулу, ни Тейваз её не содержат.(б).

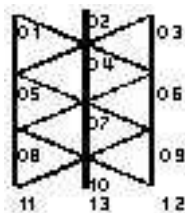


Рис. 3. Праматрица в виде неориентированного графа.

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$$

Такой способ кодировки существенно облегчает расшифровку. Вместо того, чтобы строить большую матрицу для сложного символа и проверять её на наличие комбинаций простых символов, достаточно построить алгоритм, который будет разбивать простые символы по группам, соответствующим одинаковым матриц, и позволит сравнивать матрицы между собой, отбрасывая группы матрицы, которые не соответствуют исходному коду. Для примера рассмотрим базовые символы старшего футарка. Построим иерархию базовых символов, учитывая свойство неполноты (рис.4). Сравнивая матрицы разбиения составного символа с матрицами базовых символов, осуществляем расшифровку. В базу данных передаётся информация относительно суммарного символа: координаты места, где был обнаружен предмет, время, которым определяется предмет, атрибуты предмета (назначение, материал, форма и так далее). С учётом полученной информации мы подбираем способ разбиения, и алфавит. Получив базовые символы, мы ищем при помощи языкового анализатора устойчивые и несуществующие комбинации символов, что исключит мнимые решения задачи и выделит наиболее явные. На выходе мы получим некоторое множество решений, которое можно уменьшить при помощи дополнительных условий. База знаний является накопителем начальной информации – того, что уже известно о рунах и об расшифрованных на данный момент символах, а также она хранит информацию относительно каждой последующей расшифровки, над которыми работает данная программа.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Бауэр Ф.Л., Гооз Г.* Информатика. Вводный курс. Ч.1 / Бауэр Ф.Л., Гооз Г. – М.: – Мир, 1990. – 336 с.

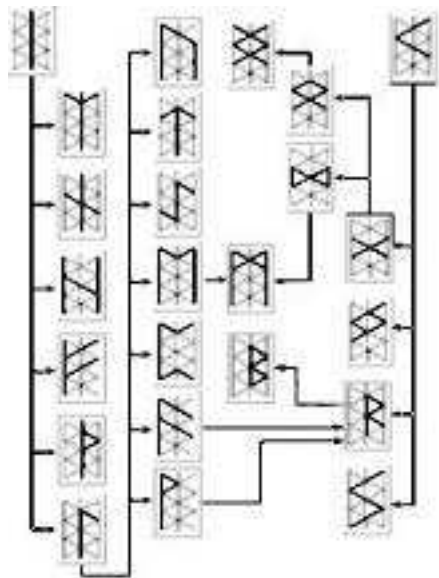


Рис. 4. Иерархия простых символов.

- [2] Хант Э. Искусственный интеллект. / Хант Э. – М.: – Мир, 1978. – 560 с.  
 [3] Колесов Е., Торстен А. Руны. Футарк классический и арманический. / Колесов Е., Торстен А. – М.: – Золотое сечение, 2007. – 264 с.  
 [4] А. Платов, А. ван Дарт. Практический курс рунического Искусства. / А. Платов, А. ван Дарт. –К.: – София, 1999. – 167 с.  
 [5] Зыков А. А. Основы теории графов. / Зыков А. А. – М.: – Наука, 1987. – 383 с.

## Коррекционные программы для детей с нарушением зрения

*Литвиненко В. О.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. ВЕРНАДСКОГО  
 ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
 КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Перед обществом поставлена благородная задача - найти пути повышения качества жизни лиц с ограниченными возможностями. В этом видят основу своей деятельности ученые и социальные работники США, Европы и России, среди которых Дж. Бернард [1], Т. А. Добровольская [2] и др.

В настоящее время более 500 млн. человек относятся к лицам с ограниченными возможностями, среди которых около 314 миллионов человек страдают

нарушениями зрения [3]. Особое внимание необходимо уделять детям с нарушением зрения, так как их эффективная реабилитация является важнейшей социальной проблемой.

В ряде тифлопедагогических и тифлопсихологических исследований показано отрицательное влияние слепоты и слабовидения на развитие познавательной деятельности, и эмоционально-волевой сферы детей (М. И. Земцова [4]; А. И. Каплан, М.С.Певзнер [5]; А. Г. Литвак [6] и др.).

Одним из эффективных средств для развития, воспитания и обучения ребенка с нарушением зрения является использование компьютерных технологий.

Целью настоящей работы является разработка адаптированных коррекционных компьютерных программ для детей с нарушением зрения.

На основании цели сформулированы следующие задачи:

- провести анализ библиографических источников для выявления наиболее эффективных коррекционных методик, не реализованных на компьютере;
- выделить необходимые требования для создания коррекционных программ;
- применить требования для создания коррекционных программ;
- осуществить экспериментальную проверку полученных коррекционных программ.

Для реализации поставленных задач была выбрана технология Macromedia Flash MX и коррекционная методика развития внимания и зрительного восприятия Григорьевой Л.П., Бернадской М.Э., Блиниковой И.В., Сонцевой О.Г. [7].

В качестве теоретических методов использовались: аналитический обзор научно-методической, психолого-педагогической и специальной литературы по проблеме исследования; анализ программ, учебных пособий и методических материалов.

Эмпирическая часть исследования реализована следующими методами: беседа, наблюдение за деятельностью детей, изучение результатов деятельности детей, опытно-поисковая работа.

После изучения статьи о людях с ограниченными возможностями зрения [8] были выделены следующие требования к компьютерным программам:

- изображения должны иметь альтернативный текст;
- количество изображений должно быть ограничено;
- необходима возможность прослушать текстовую информацию;
- необходима возможность увеличения шрифтов и картинок;

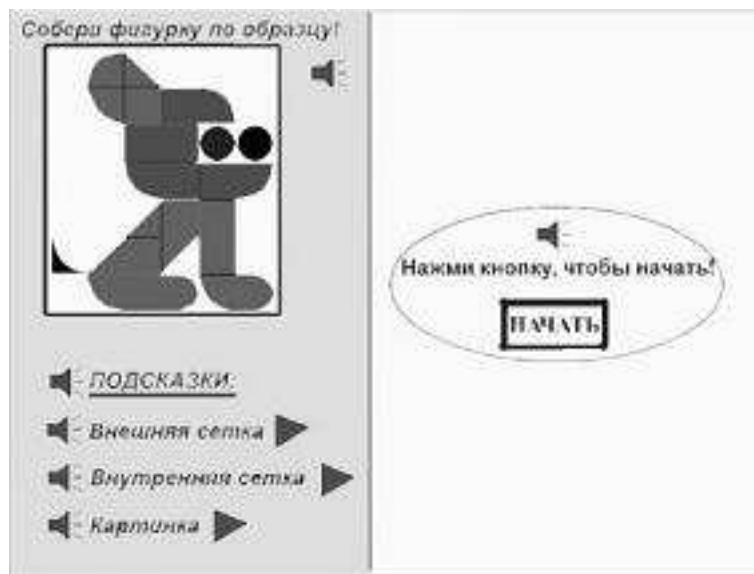


Рис. 1. Программа "Мозаика".

- контраст изображений и шрифта относительно фона должен быть высоким.

В рамках поставленной задачи и перечисленных требований разработан проект максимально дружелюбного интерфейса для коррекционных программ.

Для фона был выбран серый цвет как классический нейтральный, не отвлекающий внимания от расположенных на нем объектов [9].

Игровые элементы было решено выполнить в ярких и привлекательных для детей цветах - красном и желтом. Согласно выводам некоторых авторов, воздействие ярких цветов не раздражает детей младшего возраста, а, напротив, успокаивает, позволяя ребенку чувствовать себя комфортно [10]. Кнопки текстовых сообщений с целью активизации внимания выполнены в оранжевом цвете. Все текстовые элементы имеют звуковую альтернативу.

Экранные изображения имеют четкий (обычно черный) контур, простую для идентификации форму, достаточно крупный размер.

Изменение масштаба объектов в 2 и более раз без потери структуры и искажения содержимого обеспечивается за счет стандартных функций Flash-проигрывателя.



На рисунке 1 показан фрагмент программы для обучения анализу и синтезу сложных по структуре и числу элементов изображений.

В процессе проведения экспериментальной проверки разработанной коррекционной программы было установлено: учащиеся проявляют интерес к программе, успешно используют звуковые и текстовые подсказки, не испытывают заметных трудностей в ориентировании, успешно выполняют поставленную задачу.

Результаты проведенного тестирования свидетельствуют о корректности сформулированных требований доступности и эффективности выбранной модели для реализации коррекционных компьютерных программ.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Z. Alexander, J. Bernard. *Central Council for Education and Training in Social Work. London, 1999.*
- [2] Добровольская Т.А., Шабалина Н.Б. *Социально-психологические особенности взаимоотношений инвалидов и здоровых. Социол. исслед. -1993.*
- [3] <http://www.who.int/mediacentre/factsheets/fs282/ru/index.html>. -данные Всемирной организации здравоохранения .
- [4] Земцова М.И. *Учителю о детях с нарушениями зрения.-М.: Просвещение, 1973.*
- [5] М.И. Земцова, А.И. Каплан, М.С. Певзнер. *Дети с глубокими нарушениями зрения .-М., 1967. .*
- [6] Литвак А.Г. *Тифлопсихология. - М., 1985.*
- [7] Григорьева Л.П., Бернадская М.Э, Блинникова И.В., Сонцева О.Г. *Развитие восприятия у ребенка. Пособие для коррекционных занятий с детьми с ослабленным зрением в детском саду, начальной школе.- М.: Школьная Пресса, 2007.*
- [8] How People with Disabilities Use the Web. *Working-Group Internal Draft, 5 May 2005.*The World Wide Web Consortium (W3C)
- [9] М.Купер, А. Мэтьюз. *Язык цвета. Как использовать преимущества своего цвета для успеха в личной жизни и бизнесе - М: Изд-во ЭКСМО-Пресс, 2001.*
- [10] Базыма Б.А. *Цвет и психика. Монография. Харьков, 2001.*

## Разложение графа на трехсвязные компоненты

*Миланин А. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Ниже речь пойдет о разложении графа на трехсвязные компоненты. А именно, будут описаны: универсальный метод, упрощенный стандартный подход и эффективный алгоритм. В конце будет проведен сравнительный анализ всех методов.

## 1. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Для начала введем некоторые обозначения и определения. Пусть  $G = (V, E)$  - неориентированный граф без петель и кратных ребер (последние ограничения существенно не сказываются на работе алгоритмов, они приведены лишь в целях упрощения описания и ограждения от рассмотрения некоторых частных случаев). Две вершины  $v_1, v_2$  из  $V$  называются односвязными, если существует путь из  $v_1$  в  $v_2$  только по ребрам из  $E$ . Две вершины  $v_1$  и  $v_2$  из  $V$  называются реберно  $k$ -связными, если при удалении менее чем  $k$  ребер из множества  $E$  вершины  $v_1$  и  $v_2$  являются односвязными.  $S$  - непустое подмножество  $V$ , называется  $k$ -связной компонентой графа  $G$ , если любые две вершины  $v_1$  и  $v_2$  из  $S$  являются  $k$ -связными. Одна вершина является  $k$ -связной компонентой для любого  $k$ . Если  $S_1$  -  $k_1$ -связная,  $S_2$  -  $k_2$ -связная компоненты, и существует некоторая вершина  $v$ , принадлежащая  $S_1$  и  $S_2$ , то  $S_1 \cup S_2$  является  $\min(k_1, k_2)$ -связной компонентой. Поэтому далее нас будут интересовать только максимальные по включению компоненты.

## 2. ПЕРВЫЙ МЕТОД

Рассмотрим универсальный способ нахождения  $k$ -связных компонент в графе. Для этого укажем критерий  $k$ -связности двух вершин.  $v_1$  и  $v_2$  являются  $k$ -связными, если существует не менее  $k$  реберно непересекающихся путей из вершины  $v_1$  в  $v_2$  в графе  $G$ . Этот факт незамедлительно наталкивает на мысль о применении алгоритма нахождения максимального потока в сети [1]. Действительно, если всем ребрам приписать пропускную способность 1 и пустить поток из вершины  $v_1$  в  $v_2$ , то величина максимального потока будет отражать связность вершин. То есть для того, чтобы разложить граф на  $k$ -связные компоненты, необходимо находить максимальный поток между всеми парами вершин, и если эта величина не менее чем  $k$ , то вершины нужно объединять в компоненты.

## 3. ВТОРОЙ МЕТОД

Отыскание односвязных и двусвязных компонент - задачи, которые обладают некоторой особенностью по сравнению с общей задачей. Так, поиск односвязных компонент - задача тривиальная, решается одним применением поиска в глубину. Поиск же двусвязных компонент [2] менее очевиден, однако все же является классической задачей. Определение  $k$ -связности двух вершин можно перефразировать по-другому: две вершины  $v_1$  и  $v_2$  являются

$k$ -связными если при удалении любого одного ребра из графа,  $v_1$  и  $v_2$  являются  $(k - 1)$ -связными. Поскольку нас интересуют трехсвязные компоненты, то можно по очереди удалять каждое ребро из графа и находить двусвязные компоненты. Потом по этим результатам легко восстановить трехсвязные компоненты.

#### 4. ТРЕТИЙ МЕТОД

Перейдем к описанию разработанного автором эффективного алгоритма. Основу алгоритма составляет поиск в глубину. Напомним, что в результате работы поиска в глубину каждое ребро в графе становится ориентированным в зависимости от того, в каком направлении оно впервые было рассмотрено. А также все ребра разделяются на две группы: прямые и обратные. Прямые ребра - те, которые ведут из текущей вершины в ранее не рассмотренные. Обратные ведут из текущей вершины в некоторого предка. Вершина  $v$  имеет доступ к вершине  $u$ , если существует такой путь из вершины  $v$  в  $u$ , что он состоит из последовательности (возможно пустой) прямых ребер и заканчивается одним обратным ребром. При этом  $u$  будем называть доступной из  $v$ .

Каждой вершине в порядке обхода будет присваиваться некоторое значение - время входа  $t$ . При этом для каждой вершины мы будем определять двух ее самых ранних предков (не обязательно различных), к которым она имеет доступ. Причем на множестве еще не рассмотренных ребер должны существовать два не пересекающихся по ребрам пути к этим двум вершинам. Пусть для вершины  $v$  две ранних доступных вершины обозначаются  $l_1$  (ранняя) и  $l_2$  (вице-ранняя), тогда утверждается, что  $v$  и  $l_2$  являются трехсвязными. Даже если пропустить доказательство этого неочевидного факта, то возникает вопрос: как находить, или точнее, поддерживать информацию о вершинах  $l_1$  и  $l_2$  для каждой вершины в процессе обхода? Основная проблема заключается в удовлетворении условия о непересечении путей к  $l_1$  и  $l_2$ .

Для текущей вершины доступными будут все те, которые доступны из ее прямых сыновей плюс те, к которым ведет обратное ребро. Будем считать, что мы для каждой вершины знаем все доступные из нее и все, из которых доступна она. А в конце покажем, что эта информация не нужна. Тогда пусть есть текущая рассматриваемая вершина  $v$ . Если из нее ведет обратное ребро в  $u$ , то нужно обновить информацию об  $l_1[v]$  и  $l_2[v]$ :

*if*  $t[u] < t[l_1[v]]$  *then*  $l_2[v] = l_1[v]$ ,  $l_1[v] = u$ ;  
*else if*  $t[u] < t[l_2[v]]$  *then*  $l_2[v] = u$ ;

Однако если ребро прямое, тогда мы запускаем рекурсивный обход из вершины  $u$  и узнаем всю необходимую информацию об  $u$ . Оказывается, достаточно выбрать единственную вершину  $w$  из всего обхода, запущенного из  $u$  и через нее обновить  $l_1[v]$  и  $l_2[v]$ :

*if*  $t[l_1[w]] < t[l_1[v]]$  *then*  $l_2[v] = l_1[v]$ ,  $l_1[v] = l_1[w]$ ;

*else if*  $t[l_1[w]] < t[l_2[v]]$  *then*  $l_2[v] = l_1[w]$ ;

*if*  $t[l_2[w]] < t[l_2[v]]$  *then*  $l_2[v] = l_2[w]$ ;

При этом  $v$  должна быть доступна из  $w$ .

Этот факт требует строгого доказательства, которое здесь намеренно пропущено. Основа алгоритма на этом завершена. Утверждается, что при помощи амортизационных оптимизаций данный алгоритм можно улучшить до сложности выполнения  $O(|E|)$ . Заметим, что если вершина  $v$  стала доступной из некоторой вершины  $q$  (а именно, оказалось, что из  $q$  ведет обратное ребро в  $v$ ), то  $v$  доступна из всех вершин на пути по прямым ребрам из  $v$  в  $q$ . Это значит, что из всех вершин, из которых ведет обратное ребро в  $v$ , можно спуститься по прямым ребрам до  $v$  и выбрать лучшую вершину  $w$  на этих спусках. Через нее и надо обновлять  $l_1[v]$  и  $l_2[v]$ . Более того, в силу единственности спуска для каждой вершины  $q$ , можно для  $q$  сохранять номер лучшей вершины на спуске от нее к  $v$ . Еще важнее, что этот показатель не может ухудшиться при переходе от рассмотрения вершины  $v$  к ее некоторому предку.

Последние два замечания позволяют реализовать динамическую структуру, позволяющую поддерживать и обновлять информации о лучших вершинах  $w$  на дереве из прямых ребер. При грамотной реализации это очень сильно ускоряет работу алгоритма (собственно от сложности  $O(|V|^3)$  до  $O(|E|)$ ), скажем, кстати, что сама задача о трехсвязных компонентах представляет интерес в основном в случае  $|V| \sim |E|$ , поэтому сложность  $O(|E|) \sim O(|V|)$  существенно лучше, чем  $O(|V|^3)$ .

## 5. ЗАКЛЮЧЕНИЕ

Очевидным преимуществом первого подхода к решению является то, что им можно находить  $k$ -связные компоненты, независимо от  $k$ . Но сложность работы  $O(|V|^4)$  и огромные трудности реализации оставляют желать лучшего. Второй метод достаточно прост для реализации и имеет сложность  $O(|E|^2)$ . Тогда как авторский метод в своем лучшем варианте позволяет решать задачу за время, эквивалентное количеству введенных данных! Это означает, что для произвольных графов этот алгоритм асимптотически не может быть улучшен.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein *Introduction to Algorithms*. Massachusetts Institute of Technology, 1990. 458-476, 553-558, 567-568.
- [2] С. Окулов *Программирование в алгоритмах*. – М.: Бином. Лаборатория знаний, 2004. 154-157.

## Некоторые способы генерации поверхностей: разработка программного обеспечения

*Намашко В. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (ГРУППА 500-П)

Наглядность изучаемого материала является мощным педагогическим средством при изучении математических дисциплин. В последние годы в образовательной сфере при изучении математики все чаще возникает необходимость трехмерной визуализации объектов с помощью персонального компьютера для наглядного представления материала. Как следствие, актуальными становятся вопросы получения, хранения и обработки графических данных о сложных комплексных трехмерных объектах, будь то простейшая кривая или сложная поверхность, простая сфера или сложное человеческое тело.

В настоящее время рынок CAE (Computer Aided Engineering) приложений широко представлен такими продуктами, как Maple, Mathematica, MathCAD. Однако, данные системы, несмотря на относительную простоту управления и богатые возможности, имеют ряд недостатков. Прежде всего, решение конкретной специфической задачи в продукте общего назначения (коими являются указанные выше программные пакеты) может оказаться непрактичным и трудоемким, требовать обширного справочного материала и освоения нюансов соответствующего данным системам псевдоязыка программирования. К тому же следует учитывать, что всё это - коммерческие продукты, требующие приобретения лицензии для легального использования.

Вследствие всего вышесказанного актуальным становится создание инструментария со строго ограниченным набором возможностей (визуализация кривых и поверхностей) и дружественным пользовательским интерфейсом. Разрабатываемое приложение не должно требовать много системных ресурсов, но при этом должно обладать мощными средствами для визуализации кривых и поверхностей, задаваемых аналитическими выражениями и операциями. Приложение должно быть максимально простым в использовании и

иметь возможность записывать в локальное хранилище ранее исследованные объекты.

При реализации такого инструментария первоочередной задачей становится реализация быстрого и надежного синтаксического анализатора математических выражений (которыми описывается визуализируемый объект) для вычисления их численного значения. Наиболее оптимальным вариантом для решения поставленных задач является нисходящий левосторонний рекурсивный анализатор, скорость работы которого достигается за счет рекурсивного спуска. Кроме того, такой анализатор масштабируем (что позволяет с легкостью расширять его функциональность) и прост для реализации [1, 167-207].

В качестве графической библиотеки был выбран OpenGL. Основными аргументами, послужившими в пользу такого выбора, являются стабильность и надежность данной графической библиотеки. Невозможность работы напрямую с видеоадаптером компенсируется скоростью работы и богатым инструментарием для работы как с графическими примитивами, так и с готовыми трехмерными объектами. Также следует отметить открытость библиотеки и постоянство интерфейса.

Для реализации приложения был выбран продукт компании Borland - C++ Builder. Главным преимуществом среды разработки C++ Builder является простота создания пользовательского интерфейса. Также следует отметить, что эффективность разработки и отладки приложений достигается не только за счет использования удобных средств визуального проектирования, но и за счет высокой производительности самих компиляторов от Borland [2, 63-145].

На Рис.1 приведена схема взаимодействия основных модулей разрабатываемого программного пакета. Имеет место модульная децентрализация процесса построения геометрических объектов - будут ли эти объекты заданы с графического интерфейса пользователя или загружены с локального хранилища - перед визуализацией на экране они должны пройти синтаксический анализ и быть записаны «Диспетчером геометрических объектов» в специальные структуры, предназначенные для вывода на экран.

Для наглядности и простоты в использовании возникает необходимость добавить в разрабатываемый инструментарий возможность задания кривых и поверхностей разными способами (параметрически, явно и неявно), а также в разных системах координат (декартовых, сферических, цилиндрических, тороидальных и т.д.). Кроме того, в программный пакет входит возможность

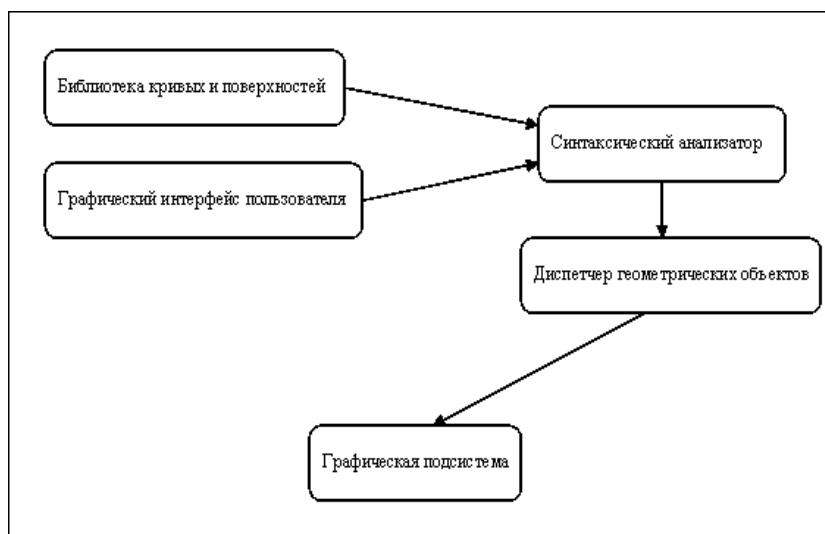


Рис. 1. Схема взаимодействия модулей приложения.

оперирования некоторыми специальными функциями, такими как тригонометрические интегралы, функции Эйри, Гамма функция, Пси функция, Эллиптически интегралы и функции Якоби. Также предусмотрена функциональность для построения плоских сечений и численного вычисления частных производных изучаемых поверхностей. А возможность полностью настроить свойства визуализируемых кривых и поверхностей, а также установить дополнительные системы освещения и цветовые параметры, направлена на то, чтобы сделать процесс работы с приложением максимально удобным.

#### СПИСОК ЛИТЕРАТУРЫ

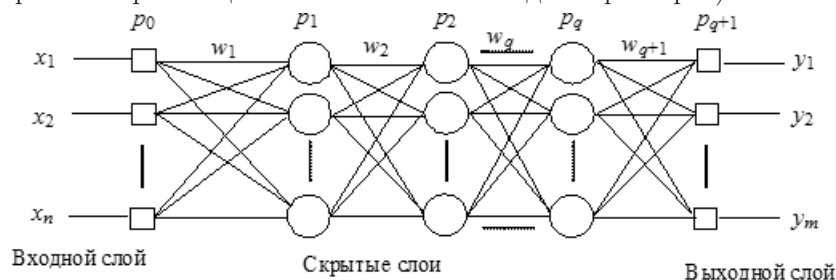
- [1] Ахо Альфред. *Компиляторы. Принципы, технологии, инструменты* / Альфред Ахо, Рави Сети, Джемфри Ульман.. – издательство "Питер 2003. - 768 с.
- [2] Архангельский Алексей. *Программирование в C++ Builder 6* / Алексей Архангельский.. – издательство Бином, - 2003. 1140 с.

## Многокритериальная оптимизация архитектуры нейронных сетей, основанная на алгоритме обратного распространения ошибки

Соседко А. Ю.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ (500-П)

Нейросетевые классификаторы — это одна из наиболее важных разновидностей искусственных нейронных сетей. Важность заключается в обширности их применения (медицинская, техническая диагностика, классификация различного рода информации и т.д.), т.е. классификаторы используются для выбора некоторых опций в зависимости от входных факторов).



На рис.5 представлена структура многослойного нейросетевого классификатора.

Здесь  $x_1, x_2, \dots, x_n$  — признаки объекта классификации, составляющие входной вектор  $x = \{x_i\}_{i=1}^n$ ;  $p_0 = n$  — число нейронных элементов в рецепторном слое;  $p_1, p_2, \dots, p_q$  — число нейронов в каждом из скрытых (обрабатывающих) слоев;  $p_{q+1} = m$  — число нейронов в выходном слое (количество классов);  $y = \{y_k\}_{k=1}^m$  — выходной вектор нейронной сети, определяющий принадлежность объекта классификации одному из  $m$  классов;  $w_1, w_2, \dots, w_q, w_{q+1}$  — векторы синаптических весов нейронной сети.

На выходе классификатора получается вектор функций активации  $y = \{y_k\}_{k=1}^m$ . Номер  $j$ , для которого выход  $y_j$  имеет максимальную активность, т.е.  $\max_{k \in [1, m]} y_k = y_j$ , соответствует номеру класса объекта классификации.

В общем виде проблема может быть формально представлена задачей

$$p^* = \arg \text{extr} Y(p) \\ p \in P$$



$Y(p)$  — целевая функция от переменного  $p$ , которое принадлежит допустимой области независимых переменных,  $extr$  — оператор экстремизации функции (max или min).

Введем несколько критериев качества функционирования нейронного классификатора. Одним критерием будет вероятность ошибки классификации. Он будет определяться экспериментально и приближенно будет вычисляться таким образом  $f_1(p) = \frac{e(p)}{N}$ , где  $f_1(p)$  — вероятность ошибки классификации  $e(p)$  — количество ошибок классификации,  $N$  — общее количество испытаний. Причем увеличение количества нейронов (в обрабатываемых слоях), в некоторых разумных пределах, приводит к уменьшению вероятности ошибки классификации. Также она должна быть ограничена некоторым заданным пределом  $f_1(p) \leq A_1$ .

Вторым критерием будет время, которое потребуется для обучения нейронной сети при некоторой заданной архитектуре  $p$ :  $f_2(p) = \sum_{k=1}^q p_k$ .

Увеличение количества нейронов приводит к увеличению времени обучения сети, т.к. данный критерий также является временем прохождения сигнала от входа нейронной сети к ее выходу. Также как и к вероятности ошибки классификации, времени задается ограничение  $f_2(p) \leq A_2$ .

Оба введенных критериев подлежат минимизации, потому что минимизируя их, выполнение поставленных задач будет проходить быстрее и с минимальными ошибками. Поэтому оператор экстремизации целевой функции приобретает вид:

$$p = \arg \underset{p \in P}{extr} Y(p) \quad , \quad \underset{p \in P}{extr} = \underset{p \in P}{min}$$

В дальнейшем будем использовать скалярную свертку критериев по нелинейной схеме компромиссов в качестве целевой функции:

$$Y(p) = Y[f(p)] = \frac{A_1}{A_1 - f_1(p)} + \frac{A_2}{A_2 - f_2(p)}$$

Где  $f(p) = \{f_r(p)\}_{r=2}^{r=1}$  — двумерный вектор частных критериев. В итоге получаем задачу оптимизации:

$$p^* = \underset{p \in P}{argmin} \left[ \frac{A_1}{A_1 - e(p)/N} + \frac{A_2}{A_2 - \sum_{k=1}^q p_k} \right]$$

В данной формуле для нахождения  $e(p)$  должна быть использована некоторая экспериментальная процедура.

Предложенный подход был реализован в виде программы, позволяющей моделировать процесс обучения нейронной сети обратного распространения ошибки и оптимизацию ее архитектуры с помощью использования скалярной свертки по нелинейной схеме компромиссов.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Головки В.А. Нейронные сети: обучение, организация и применение. - М.: ИПРЖР, 2001. - 256 с.
- [2] Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение и применение. - Харьков: ТЕЛТЕХ, 2004. - 372 с.

## Секция ИНФОРМАТИКИ

### Разработка поисковой системы на основе латентно-семантического индексирования

*Абдулхаиров А. Р.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Информация сегодня – один из самых ценных ресурсов во всех сферах человеческой жизни. Однако память ограничена, а количество информации в разных формах растет с каждым днем. В связи с этим становится актуальной задача поиска нужной информации в разного рода источниках. Пожалуй, наибольшую трудность представляет поиск текстовой информации – человеческий язык достаточно неоднозначен и часто допускает различные представления одинаковых по смыслу сообщений.

Основным способом поиска текстовой информации в интернете является поиск по ключевым словам, однако, как показывает практика, такой поиск зачастую дает много «шума». Актуален вопрос релевантности – соответствия между желаемой и действительно получаемой информацией, а также возможность использования в качестве поискового запроса естественно-языковых конструкций. Разрабатываемая в рамках дипломной работы поисковая система состоит из нескольких программных модулей и имеет ряд ключевых этапов для осуществления релевантного поиска по исходному запросу.

Первым этапом работы предлагаемой поисковой системы является сканирование страниц в сети интернет, составляющих в дальнейшем пространство поиска документов. Для перехода с одной Web-страницы на другую и выделения списка слов документа используется специальный программный модуль (HTML-сканер). При посещении HTML-сканером web-страницы происходит разбор документа на «полезный контент», «Web-информацию» и «поисковую информацию». Полезный контент – это непосредственно смысловой текст документа. «Web-информация» – некая дополнительная информация, которая подчеркивает некие особенности документа, такие как заголовки, описания и т.д. и т.п. Поисковая информация – это информация, поступающая отдельно и характеризующая поисковый трафик для данного документа.

Извлеченная из web-страницы полезная информация сохраняется, и в дальнейшем используется для индексирования. По существующим на отсканированной странице ссылкам программный модуль переходит на следующий Web-документ и продельывает разбор контекста уже для него. Таким образом для каждого документа накапливается набор употребляемых в нём слов. Большую трудность на этом этапе представляет то, что сайты имеют разную структуру и соответственно разную значимость контента, также необходимо отсеивать «информацию-мусор», которая в дальнейшем будет негативно влиять на результат индексирования.

Для эффективного использования латентно-семантического индексирования в разрабатываемой поисковой системе осуществляется анализ и преобразование собранного лексикона по следующим правилам:

- Удаление всех элементов форматирования и пунктуации.
- Разбиение контента на предложения и выделение слов из предложений.
- Исключение предлогов, местоимений, вводных слов, а также слов присутствующих во всех документах и слов, которые имеются только в единичных документах.
- Лемматизация слов(приведение слов к основной, исходной форме).

На основе измененного лексикона формируются пары Слова-Документы.

Следующим этапом работы поисковой системы является Латентное Семантическое Индексирование (Latent Semantic Indexing или LSI) – это способ, который поисковые системы используют для просмотра и ранжирования страниц сайтов в более естественной, или человеческой манере. Основная идея латентно-семантического анализа заключается в том, что совокупность всех контекстов, в которых встречается и не встречается данное слово, задает множество обоюдных ограничений, которые позволяют определить похожесть смысловых значений слов и множеств слов между собой. Исходной информацией для LSA является матрица слов на документы, которая описывает используемый для обучения системы набор данных. Элементы этой матрицы содержат частоты использования каждого слова в каждом документе. Используя разложение исходной матрицы по сингулярным значениям (SVD – Singular-Value Decomposition), большая исходная матрица разлагается во множество из  $k$  ортогональных матриц, линейная комбинация которых является хорошим приближением исходной матрицы. Идея такого разложения и суть латентно-семантического анализа заключается в том, что если использовать

теорему о сингулярном разложении матриц, и в качестве исходной матрицы использовать матрицу слова-документы, то матрица, содержащая только  $k$  первых линейно независимых компонент исходной матрицы, отражает основную структуру ассоциативных зависимостей, присутствующих в исходной матрице, и в то же время не содержит шума. Таким образом, каждое слово и документ представляются при помощи векторов в общем пространстве размерности  $k$ . Близость между любой комбинацией слов или документов может быть легко вычислена при помощи скалярного произведения векторов. Этап индексирования является самым трудоёмким процессом так как разложение матриц и сохранение сингулярных значений для слов и документов требуют большого количества вычислений и обращений к серверу БД. Очень важным моментом работы ЛСА является построение «обратного индекса» на основе полученных сингулярных значений – способа однозначно описать документ в пространстве поиска так, чтобы максимально быстро в дальнейшем извлечь его из базы данных.

Заключительным этапом работы предложенной поисковой системы является обработка запросов и поиск наиболее релевантных этому запросу документов. Для этого поисковой запрос разбивается на слова из лексикона, затем в проиндексированной базе данных, используя ранее найденные сингулярные значения, для этих слов выбираются наиболее близкие по значению документы. Таким образом, пользователь получает ранжированный список документов, наиболее близких по смыслу введенному поисковому запросу.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Edel Garcia*. LSI Keyword Research A Fast Track Tutorial.
- [2] *W. Kintsch, A. Bowles* Metaphor comprehension: What makes a metaphor difficult to understand? // *Metaphor and Symbol*, 2002, № 17. – 249-262.
- [3] *А.Ю. Пастухов, П.В. Сердюков, Н.А. Козлова, С.А. Чернов*. Латентный семантический анализ: новые решения в Internet. // *Информационные технологии*, 2001, № 6.

## Распознавание печатных и рукопечатных символов

*Буклерская Е. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Оптическое распознавание текста (OCR) — механический или электронный перевод изображений рукописного, рукопечатного или печатного текста (полученного при помощи сканера) в программно редактируемый текст. Распознавание символов применяется для приведения печатной продукции к электронному виду. OCR — область научных исследований на границе распознавания образов, искусственного интеллекта и компьютерного зрения. Изначально оптическое (с использованием оптических приемов и технологий, таких, как зеркала и линзы) и цифровое распознавание символов считались различными областями. Но сейчас термин «OCR» охватывает более широкий круг понятий, включая в себя также и цифровое распознавание.

Распознавание латинских символов в печатном тексте считается по большей части решенной задачей для приложений, работающих с «чистыми» изображениями высокого качества. Обычно точность распознавания для таких документов достигает 99% [1]. Стопроцентная точность может быть достигнута только с помощью последующей проверки человеком. Однако, следует учитывать, что проблема оптимального синтеза уже существующих алгоритмов для создания более эффективной, быстродействующей и многофункциональной системы, остается актуальной. Кроме того, такие задачи, как распознавание рукопечатных и написанных прописью символов, а также символов других (не латинских) шрифтов, до сих пор является предметом активных исследований.

**Целью** работы является программная реализация системы, способной распознавать печатные и рукопечатные символы. В качестве подхода к решению задачи распознавания был выбран метод, основанный на векторизации.

Реализация поставленной цели потребовала решения следующих задач:

- изучить текущее состояние научной области, занимающейся распознаванием символов;
- ознакомиться и сравнить существующие подходы к созданию распознающих систем;
- сделать краткий обзор существующего программного обеспечения и рассмотреть технологии, использованные для его создания;
- найти наиболее эффективный алгоритм распознавания или совместить несколько для его получения;
- ознакомиться со способами улучшения качества распознавания.

Работа носит научно-исследовательский характер и состоит из трех основных разделов. Первый раздел содержит обзор современной литературы по теме магистерской работы. Введено понятие оптического распознавания

символов; описаны существующие подходы и алгоритмы, используемые для решения задачи распознавания; рассмотрены проблемы, с которыми может столкнуться пользователь на начальных этапах работы с распознаваемым документом. Также приведен обзор существующего программного обеспечения, эффективно решающего задачу распознавания. Дополнительно дано понятие векторизации растрового изображения и рассмотрены некоторые наиболее известные алгоритмы автоматической векторизации бинарных растров.

Второй раздел посвящен описанию процесса распознавания символов с использованием алгоритма векторизации. Рассмотрены этапы подготовки изображения к распознаванию. Детально описаны алгоритмы нахождения контуров изображения и применяемый для распознавания алгоритм векторизации на основе триангуляции. Его результатом является векторизованный скелет символа. Рассмотрены методы улучшения качества полученного результата, основанные на уменьшении количества вершин в скелете (включая метод удаления ложных ответвлений) [2]. Дополнительно приведен подробный алгоритм генерализации ломаной, основанный на методе наименьших квадратов.

В третьем разделе приводится программная реализация процесса распознавания печатных и рукопечатных символов на основе структурного алгоритма. Программно реализован алгоритм распознавания отдельных рукопечатных символов латинского алфавита, подаваемых на вход либо в качестве существующего изображения формата jpg (256x256 пикселей), либо задаваемых пользователем с помощью области для рисования. Программная реализация включает в себя следующие этапы.

1. **Выделение символа** на изображении и преобразование бинарного растра в матрицу, содержащую единицы на месте черных пикселей, и нули - на месте белых. Также на этом этапе происходит дозамыкание линий и заполнение разрывов.
2. **Скелетизация** (уточнение) полученного символа с помощью модифицированного алгоритма Зонга-Суня. На вход подается матрица символа, состоящая из нулей и единиц, на выходе образуется двумерный массив, представляющий собой цепочки пикселей, составляющих скелет. Формирование цепочек происходит следующим образом: совершается обход скелета по часовой стрелке, если встреченный черный пиксел имеет только одного черного соседа, то он заносится в текущую строку двумерного массива, и рассматривается его сосед. Если же пиксел "развилочный" то есть имеет нескольких соседей, он также заносится в текущую строку, после чего создается новая строка, в которую он

заносится в качестве первого элемента, затем осуществляется переход на первого соседа (относительно обхода по часовой стрелке).

3. **Оптимизация** полученного скелета. На этом этапе удаляются все лишние строки матрицы - содержащие единичные элементы, не представляющие важности для распознавания, и содержащие пары соседних точек (которые поглощаются более длинными отрезками, после чего удаляются).
4. **Аппроксимация** полученных цепочек. После этого этапа каждая из строк двумерного массива будет содержать только важные для распознавания пиксели - вершины графа, представляющего символ.
5. **Распознавание**. Первоначальный алгоритм распознавания основан на анализе степеней вершин, полученных после аппроксимации, а также некоторых признаков отдельных букв. Программа с достаточной точностью распознает большую часть рукопечатных символов, подаваемых на вход, допуская перекосы изображения, кривизну линий и незамкнутые контуры.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Травин А.* Технологии оптического распознавания текстов // Электронный офис. — 1996.
- [2] *Optical character recognition.* — [http://en.wikipedia.org/wiki/Optical\\_character\\_recognition](http://en.wikipedia.org/wiki/Optical_character_recognition).
- [3] *Чертов А. А.* Векторизация бинарных растров на основе триангуляции и алгоритмы постобработки. — Томск, 2006. — 41 с.

## Дискретное извлечение корня

*Винц А. Б.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

### 1. ВВЕДЕНИЕ

Теория чисел (в частности, её подраздел - модульная арифметика) получила в последние несколько лет огромное распространение. Стало известно, что операции, выполненные по простому модулю, являются устойчивыми к взлому способами кодирования чисел; соответственно, появилась необходимость в эффективном и правильном выполнении этих операций на современных ЭВМ.



Данная статья посвящена рассмотрению самой трудоёмкой с точки зрения машинных вычислений операции модульной арифметики - дискретному извлечению корня.

Постановка задачи дискретного извлечения корня такова:

Дано простое число  $p$ , целые числа  $a \in [0 \dots p - 2]$ ,  $b \in [0 \dots p - 1]$ . Найти такое целое число  $x \in [0 \dots p - 1]$ , что

$$x^a = b(p)$$

Если искать решение задачи полным перебором по всем вариантам ответа, то оценка сложности подобного алгоритма будет  $O(p \cdot \log_2 a)$  (перебор всех  $x$  за  $O(p)$  операций с последующим возведением  $x$  в степень  $a$  за  $O(\log_2 a)$  операций. Таким образом, задача дискретного извлечения корня оказывается сложнее задачи дискретного логарифмирования, решение которой полным перебором по вариантам ответа даёт лишь  $O(p)$  операций.

Следовательно, необходим алгоритм, находящий дискретный корень из числа намного быстрее, чем за  $O(p)$  операций.

Предлагается решение, сводящее задачу дискретного извлечения корня к задаче дискретного логарифмирования. Известно, что по простому модулю  $p$  существует первообразный корень  $g$ . Вследствие этого можно сделать подстановку  $x = g^y$ , приводящую уже к задаче

$$g^{ay} = b(p),$$

где неизвестное  $y$  стоит уже в показателе степени. Таким образом, исходная задача сводится к задаче дискретного логарифмирования с предшествующей ей задаче нахождения первообразного корня по модулю  $p$ .

## 2. ЗАДАЧА НАХОЖДЕНИЯ ПЕРВООБРАЗНОГО КОРНЯ

**Теорема 1.** Если  $g$  - первообразный корень по модулю  $p$ , то  $g^{\frac{p-1}{2}} = p - 1(p)$

*Доказательство.* Действительно, согласно теореме [1], все числа по простому модулю  $p$  в степени  $\frac{p-1}{2}$  дают либо 1, либо  $p - 1$ . Однако первообразный корень в любой положительной степени, не превышающей  $p - 2$ , не может принимать значение 1 по модулю  $p$ . Таким образом, для первообразного корня  $g: g^{\frac{p-1}{2}} = p - 1(p)$ , что и требовалось доказать.  $\square$

**Теорема 2.**  $g$  не является первообразным корнем по простому модулю  $p$  тогда и только тогда, когда  $\exists a \in [1 \dots p - 2], p - 1 : a :$

$$g^a = 1(p)$$

*Доказательство.* Достаточность очевидна. Докажем необходимость.

Так как  $a$  не является первообразным корнем, то  $\exists a_1, a_2 \in [0 \dots p-2]$ :

$$g^{a_1} = g^{a_2}(p)$$

Тогда, взяв  $a = (a_1 - a_2) \text{ MOD } (p-1)$ , получим  $g^a = 1(p)$

Возьмём минимальное из таких  $a$  и докажем, что  $p-1 \mid a$ . Действительно, пусть  $p-1 = a \cdot q + r$ , где  $r = (p-1) \text{ MOD } a$ . Тогда  $g^{p-1} = (g^a)^q \cdot g^r = 1^q \cdot g^r = g^r$ .

Так как  $\forall g \in [1 \dots p-1] : g^{p-1} = 1(p)$ , то и  $g^r = 1$ . Однако  $r$ , являющийся остатком от деления  $p-1$  на  $a$ , меньше  $a$ . Следовательно  $r = 0$ . Теорема доказана.  $\square$

Обозначим  $\min_{a \in [1 \dots p-2]} a : g^a = 1$  через  $a_g$ . Ясно, что  $\forall g$ , не являющихся первообразным корнем:  $g^{\frac{a_g}{2}} = p-1(p)$ . А тогда все степени вида  $g^{(2k+1) \cdot \frac{a_g}{2}} = p-1(p)$  при целом неотрицательном  $k$ .

**Теорема 3.**  $g$  является первообразным корнем по простому модулю  $p$  тогда и только тогда, когда для любого простого  $x > 2$ ,  $\frac{p-1}{2} : x : g^{\frac{p-1}{2x}} \neq p-1(p)$

*Доказательство.* Необходимость очевидна. Докажем достаточность от противного.

Действительно, пусть  $g$  - не первообразный корень по модулю  $p$ . Тогда  $\forall k : g^{(2k+1) \cdot \frac{a_g}{2}} = p-1(p)$ . Обозначим  $q = \frac{p-1}{a_g}$ . Оно нечётно, так как в противном случае  $g^{\frac{p-1}{2}} = g^{q \cdot \frac{a_g}{2}} = (p-1)^q$  было бы  $1(p)$ . Пусть  $x$  - минимальный простой делитель  $q$ . Тогда  $g^{\frac{q \cdot a_g}{2x}} = p-1(p)$ . Но  $g^{\frac{q \cdot a_g}{2x}} = g^{\frac{p-1}{2x}}(p)$ . Мы пришли к противоречию.  $\square$

Последняя теорема даёт алгоритм проверки некоторого числа  $g$  на то, является ли оно первообразным корнем по модулю  $p$ . Перебирая все  $x$  - простые делители числа  $\frac{p-1}{2}$ , не равные 2 и возводя  $g$  в степень  $\frac{p-1}{2x}$ , необходимо проверить результат на равенство  $p-1$  по модулю  $p$ . Если такие равенства не имели место, то  $g$  - первообразный корень по модулю  $p$ . Сложность такой проверки для одного числа -  $O(\sqrt{p} \cdot \log_2 p)$ .

Как показывает практика, первообразные корни по модулю  $p < 1000000000$  - сравнительно небольшие числа. Известна оценка Шуупа, согласно которой  $g = O(\log_2^6 p)$ .

### 3. ДИСКРЕТНОЕ ЛОГАРИФИРОВАНИЕ

Существует алгоритм, решающий задачу дискретного логарифмирования за  $O(\sqrt{p} \cdot \log_2 p)$  операций. Ниже приведено его краткое описание.

(Напомним, что решается задача  $g^y = b(p)$ )

1.  $q := \lfloor \sqrt{p} \rfloor$
2. *for*  $i := 0$  *to*  $q$  *do*  $ar[i] := g^{i \cdot q}$
3.  $sort(ar)$
4. *for*  $i := 0$  *to*  $q$  *do*
5. *if*  $\frac{b}{g^i} \in ar$  *then*  $z := q \cdot ( ar, \frac{b}{g^i} ) + i$

Заметим, что так как  $g$  - первообразный корень по модулю  $p$ , то решение задачи дискретного логарифмирования всегда существует. Однако решение задачи дискретного извлечения корня существует тогда и только тогда, когда существует решение задачи  $y \cdot a = z(p - 1)$ . Так как  $p - 1$ , вообще говоря, не является простым, то нахождение обратного элемента по модулю  $p - 1$  - задача не всегда разрешимая. Если же решение существует, то число  $x = g^y$  и будет решением задачи дискретного извлечения корня.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ, 2-е издание. // Издательский дом "Вильямс 2007 С. 968–973.

## Monotonicity testing of boolean functions

*Dzheialova A. I.*

TAURIDA NATIONAL V. VERNADSKY UNIVERSITY  
FACULTY OF MATHEMATICS AND INFORMATICS  
CHAIR OF INFORMATICS (GROUP 500-I)

#### INTRODUCTION

The problem of testing comes up in the study of self-testing and self-correcting programs. This in turn has applications to probabilistically checkable proofs. Probabilistically checkable proofs are especially important because they have been shown to yield an alternative characterization of the complexity class NP and this in turn has led to proofs of the intractability of approximation problems. Checking a function P for correctness can be done in two phases. The first tests whether P has a given property. If so, the second phase then checks if P is indeed the correct function, using a test that depends on the property holding. Testing whether a function is a low-degree polynomial and well-studied problems.

## 1. MONOTONICITY TESTING

We want to know what properties hold of a function. Recall that we seek a local test for a global property, that is, membership in some family  $\mathcal{F}$ . A characterization to the extent that they yield tests.

There is a monotonicity test of complexity  $\text{poly}(n/\epsilon)$ . This test samples the function at arguments of its choice.

The test selects a random  $x \in \{0, 1\}^n$  and  $i \in 1, \dots, n$ . We obtain  $y$  from  $x$  by flipping the  $i$ th bit in  $x$ . The test rejects if  $x, y$  is a monotonicity-violating tuple. Otherwise the steps above are repeated up to  $n^3/\epsilon$  times. Let  $\delta_\mu(M)$  denote the fraction of such pairs  $(x, y)$  which violate monotonicity for a function  $f$ .

**Definition 1.** Let  $\epsilon(f)$  be the distance of  $f$  to monotone functions.

The main result is:

$$\delta_\mu(f) \geq \epsilon(f)/n^3$$

from which the correctness of the test follows.

A test for membership in  $\mathcal{F}$  is an implementation of a robust exact local characterization of  $\mathcal{F}$ . A test must accept a function  $f$  with high probability if  $f$  is  $\epsilon$ -far from  $\mathcal{F}$ . Otherwise, the test may accept or reject  $f$  arbitrarily.

## CONCLUSION

We are interested in characterizations which are not only local, exact and robust, but also efficient. The complexities of the tests are in all cases depend on  $\epsilon$ , the maximum distance beyond which a family and a putative member are no longer considered "close". The complexity of monotonicity tests are also dependent on  $n$ , the arity of the given function. We are interested in characterizations (tests) with complexity polynomial in  $n$ , the arity of the given function. We are interested in characterizations (tests) with complexity polynomial in  $n$ , for any  $n$ -adic Boolean function to be tested.

## REFERENCES

- [1] Krzysztof Michał Majewski B.Sc., McGill *Probabilistic testing of Boolean functions*. – University, 1998.
- [2] Oded Goldreich *Testing Monotonicity*. – Dept. of Computer Science and Applied Mathematics, 1999.

# Система Автоматического Манипулирования Сайтами

Гусаров А. А.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Мы живем в эпоху правления интернета. Который проникает в каждый аспект нашей жизни. На нем завязаны взаимодействия огромных компаний и будущее мелких предприятий. Это возможность каждого услышать и быть услышанным. Реализовать свой потенциал и посмотреть на потенциал других. Интернет - это огромный и невероятно быстро развивающийся организм. Естественно при таких обстоятельствах появление огромного количества продуктов связанных с нетом - было предсказуемо и ожидаемо.

В связи с повторяемостью ежедневных операций, производимых обыкновенным пользователем, автоматизация процесса сёрфинга - есть задача со всех сторон интересная и востребованная. *Основная идея состоит в том, чтобы предоставить пользователю виртуальный браузер, через который он и будет совершать свои действия с сайтом при помощи псевдоязыка программирования, придуманного и реализованного специально для этого браузера и позволяющего написать простой алгоритм, совершаемых пользователем действий, запомнить его и, при надобности, повторить.*

Т.к. происходит непосредственная работа с интернет-контентом, то естественное браузер реализуется тоже через интернет-ориентированные инструменты: на языке PHP на базе фреймворка Symfony 1.4 [1] с использованием базы данных Mysql 5 [2]. Используя паттерн программирования MVC [3], мы создаем первоначальный каркас, на который будет "навешиваться" нужный нам функционал. Каркас - это портал, с возможностью работы с пользователями (авторизация - распознавание пользователя в системе), работы с статьями (добавление/редактирование) и интегрированным ORM (используется Doctrine 1.2 [4]). После написания каркаса - под него пишутся специальные плагины, реализующие поставленную выше цель. Пишутся, в основном на PHP, но для фоновых задач (Cron) используется Perl.

Т.к. происходит непосредственная работа с интернет-контентом, то естественное браузер реализуется тоже через интернет-ориентированные инструменты: на языке PHP на базе фреймворка Symfony 1.4 [1] с использованием базы данных Mysql 5 [2]. Используя паттерн программирования MVC [3], мы создаем первоначальный каркас, на который будет "навешиваться" нужный

нам функционал. Каркас - это портал, с возможностью работы с пользователями (авторизация - распознавание пользователя в системе), работы с статьями (добавление/редактирование) и интегрированным ORM (используется Doctrine 1.2 [4]). После написания каркаса - под него пишутся специальные плагины, реализующие поставленную выше цель. Пишутся, в основном на PHP, но для фоновых задач (Cron) используется Perl.

Перед тем, как дальше углубляться в механику работы, следует разобраться в некоторых аспектах современных сайтов, для более полного понимания процесса взаимодействия Пользователь ' Браузер ' Сайт. В последнее время стали очень популярны, так называемые, CMS (Система управления содержимым (контентом), от англ. Content management system, CMS [5]) - компьютерная программа или система, используемая для обеспечения и организации совместного процесса создания, редактирования и управления текстовыми и мультимедиа документами (содержимым или контентом). Обычно это содержимое рассматривается как неструктурированные данные предметной задачи в противоположность структурированным данным, обычно находящимися под управлением СУБД. Таких вот CMS сейчас немалое множество, одни из самых популярных - это DRUPAL, Joomla, Bitrix, Wordpress и другие. В связи с огромным количеством личных дневников-блогов, соответственно стали появляться и движки, которые предназначены именно для них. Нельзя обойти стороной и Wiki-движок, который является довольно ярким и весомым событием в развитии ВЕБ 2.0. Огромная Википедия - яркий показатель мощности этого продукта. Естественно для каждого сайта, созданного или с нуля или с помощью одной из CMS или других инструментов - используется верстка CSS - технология описания внешнего вида документа, написанного языком разметки.

Итак, чтобы пользователь мог программно взаимодействовать с сайтом, нам необходимо получить абсолютно всю информацию со страницы: все теги, все атрибуты тегов, весь контент, вся техническая неотображаемая информация, все стили и все Java-скрипты. После распарсивания сайта на отдельные элементы, построения его структуры и идентификации блоков информации - выводим пользователю дополненный "браузерными" элементами сайт. Пользователь имеет возможность видеть структурные блоки сайта, отфильтрованный от "нежелательных элементов" сайт (как-то баннеры, скрипты и прочее). Далее идет черед написания алгоритма действий. С помощью специального языка имеется возможность описывать последовательность действий, исходя из предоставленных блоков (каждому блоку присвоен свой ID для взаимодействия с ним на уровне кода). После валидации полученного запроса,

происходит его сохранение в базу, для дальнейшей работы. Пользователь может в дальнейшем вернуться к своему алгоритму, повторить его или модифицировать. Благодаря использованию фреймворка для разработки портала - приложение можно расширять и добавлять в него новый функционал.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Getting started with Symfony 1.4* –  
[http://www.symfony-project.org/getting-started/1\\_4/en/](http://www.symfony-project.org/getting-started/1_4/en/)
- [2] *Getting started with MySQL* –  
[http://dev.mysql.com/tech-resources/articles/mysql\\_intro.html](http://dev.mysql.com/tech-resources/articles/mysql_intro.html)
- [3] *Википедия Model-View-Controller* –  
<http://ru.wikipedia.org/wiki/Model-View-Controller>
- [4] *Doctrine 1.2 Documentation* –  
[http://www.doctrine-project.org/documentation/manual/1\\_2/en/introduction](http://www.doctrine-project.org/documentation/manual/1_2/en/introduction)
- [5] *Википедия CMS* –  
[http://ru.wikipedia.org/wiki/Система\\_управления\\_содержимым](http://ru.wikipedia.org/wiki/Система_управления_содержимым)
- [6] *PHP official site* – <http://www.php.net>

## Some of statements and objectives of the agent-oriented modeling

*Koval A.*

TAURIDA NATIONAL V. VERNADSKY UNIVERSITY  
FACULTY OF MATHEMATICS AND INFORMATICS  
CHAIR OF INFORMATICS (GROUP 401-I)

*This article deals with basic definitions of the AI agent-oriented modeling theory. Examples of the applications based on this theory are given. Usage of agents in the game development are described.*

#### INTRODUCTION

Theory of the intelligent agent modeling is an agent-oriented approach to the description of the artificial intelligence. The basic paradigms of this modern advanced approach are [1]:

— the *intellect* must be calculable subpart of the general system. It must provide successful achievement of the goals;

— the model is an *intelligent agent* if three submodels are distinguished: *sensors* for data perception from informative space; *decision making module*, responsible for return action generation; *actuators* with the ability to affect informative space in different ways;

— general part of calculation abilities of the model is dedicated to the realization of those methods and algorithms which provides most optimal movement in the informative space (such as the algorithms of the decision making and optimal solutions search).

#### 1. BASIC DEFINITIONS OF AGENT-ORIENTED MODELING OF THE ARTIFICIAL INTELLIGENCE AND CLASSIFICATION OF THE INTELLIGENT AGENTS

**Definition 1.** *The object with the abilities of data perception through sensors and return action implementation through actuators can be named as agent [1].*

**Definition 2.** *Everything collected by an agent through sensors during the lifetime can be named as the sequence of the perception acts [1].*

**Definition 3.** *The function of the agent is the projection of any chosen sequence of the perception acts onto the set of return actions [1].*

**Definition 4.** *The rational agent is the agent with the ability to find an optimal solution represented by the set of return acts for any given sequence of the perception acts; this set of return acts must be best in the sense of the performance index maximization considering all the facts given by this sequence and built-in predefined knowledge base of the agent [1].*

**Definition 5.** *The intelligent agent is the rational agent with goal-oriented functioning and strong reflection abilities [1].*

Classes of the agents [1]:

1. *Reflex-based agents* can operate only on the basis of the current knowledge. The function of the agent is presented as "if — then" knowledge base. Can operate only with the fully observed environment

2. *Model-based agents* operate on the basis of the world model. This agents can store a snapshot of some environmental part which is currently out of view. This ability allows them to operate in the partially observed environment.

3. *Goal-based agents* operate on the basis of the priority stack of the most desirable situations. Due to this agents are able of optimal actions generation.

4. *Utility-based agents* operate on the basis of the *utility function*. This functions projects a set of states onto the set of state's usefulness measures.

5. *Learning agents (Autonomous intelligent agents)*: this is the most used class of agents. They are independent in the sense of the self-education ability and the ability to adopt for the regularly changing environment. The key feature of this agents is the *reflection* — the ability to analyze and modify oneself actions according to the built-in predefined *performance standard*.



## 2. APPLICATIONS OF AGENT-ORIENTED MODELING OF THE ARTIFICIAL INTELLIGENCE

*Price comparison service:* agents surfe the internet to gather information about products and services, save the history of purchases, monitoring the seasonal activity of buyers and so on [2]. Example — <http://www.amazon.com>.

*Personal intelligent agent:* agents represent a complex system that performs the functions of "assistant". Agent performs various tasks such as searching the web, news gathering; it can communicate on a particular topic or assist in the computer game etc. [3].

*Monitoring and surveillance agents:* agents monitor the status of inventory in the facility, manage the planning, conduct surveillance of computer network. Example — NASA's Jet Propulsion Laboratory Agent [4].

*Data mining agents:* agents process data for later use. They solve the problems of classification, pattern recognition, prediction and other problems [4].

*Artificial intelligence in computer games:* agents are all objects of a computer game that can make decisions, perform moves, compete with each other. Agent-oriented modeling is used in this area not only because of the obvious connection with the AI, but also because it is a ready-to-use design pattern. This is allows reduce the cost of software development [4].

## 3. INTELLIGENT AGENT DEVELOPMENT FOR COMPUTER GAMES

It is proposed to develop an autonomous agent for a computer game.

*Short game description:* the application is a model of a military conflict. Agents model the AI of the autonomous combat robots of warring parties. Objectives of each party are different — from the total annihilation of the enemy to the capture-and-hold of the strategic resources.

Stages of the intelligence agent model development [1]:

1. *Description of environmental properties.* The environment is partially observed, strategic, sequential (coherent), dynamic, continuous, multi-agent.

2. *Agent model selection.* Agent model selection is usually based on the description of the environment. In this case it is an autonomous agent.

3. *Algorithms selection.* At this stage developer selects algorithms for data filtering, decision-making, performance monitoring etc. and gives the model predefined knowledge base and rules of learning.

4. *Standalone operation.* The model is implemented in a software solution and the agent starts to act *autonomously*. It actively interacts with other agents

and accumulates the knowledge base. Due to the reflection agent implements assessment and optimization of itself actions.

A special database with a decision-tree structure accumulates all of cause-effect chains "action of the agent — the reaction of the environment — reflection". Database is the field of actions for specially developed super agent. This agent estimates the database from time to time. It selects the most optimal solutions and generates guidance instructions to subagents. This instructions change the knowledge base of subagents so that best chains of solutions can be instanced more frequently.

Algorithms selection is the most important part of whole modeling process. State of the art publications and approaches [2], [3] stresses on such families of algorithms as neural networks, fuzzy logic, genetic evolution. Graph theory with such instantiations as decision trees, A-star pathfinding, flow calculation is the classical alternative.

The innovative usage of reflection — fundamental part of human psychology — in the agent architecture is the key to construction of so-called "human-like behavior". This (with usage of cutting edge 3D technologies) allows to create a very realistic games. These games can be very competitive in the sense of strong intellectual opposition.

Last, but not least, technique of agent modeling is the usage of subagents. This idea was borrowed from nature, where ants colonies and fish flocks can be regarded as super agents systems consisting of subagents. It must be noted that the very idea of hierarchical agent architecture is more often used for description of flock-like formations, like tactical squad of units in computer games. This gives developers very handy opportunity for advanced generalized programming usage in it's original sense.

#### CONCLUSION

The considered intelligence agent modeling approach is a generalization of most advanced researches in this direction. This is constantly developing and relatively young theory. Participation in the development of this theory of such companies as Google, NASA, Microsoft, Amazon.com, EA, Ubisoft, BioWare [4] emphasizes relevance of this trend.

#### REFERENCES

- [1] Russell S., Norvig P. *Artificial Intelligence: A Modern Approach (3rd Edition)*. – New Jersey: Prentice Hall, Upper Saddle River 07458, 2009. –1152 p.

- [2] Hagrais H., Sobh T. *Intelligent Learning and Control of Autonomous Robotic Agents Operating in Unstructured Environments*. – <https://www.bridgeport.edu/pages/1.asp>
- [3] V. Florian R. *Autonomous artificial intelligent agents*. – Center for Cognitive and Neural Studies (Coneural), Technical Report Coneural-03-01, February 4, 2003
- [4] MIT *Agents*. – [www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/Agents](http://www.aaai.org/AITopics/pmwiki/pmwiki.php/AITopics/Agents)

## Анализ принципов и критериев ранжирования в современных поисковых системах

*Коринев М. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Последнее десятилетие развития информационных технологий характеризуется все более интенсивным использованием информационных ресурсов системы Интернет. Количество информации в сети постоянно увеличивается, поэтому все более актуальными являются эффективные алгоритмы поиска, систематизации и оценивания значимости электронных документов в сети Интернет используются поисковые алгоритмы ранжирования и не только. Актуальность решения проблем построения поисковых систем определяется все более высокими требованиями, предъявляемыми к алгоритмам упорядочивания и выделения наиболее важной информации в просторах Интернет.

Целью данной работы является анализ основных критериев поиска и способов ранжирования информации в глобальной сети.

Анализируя принципы построения программных продуктов, использующих большие вычислительные мощности, можно выделить основные требования для организации эффективной поисковой системы: минимальные вычислительные ресурсы, в связи с огромным количеством обрабатываемой информации, наиболее достоверное ранжирование информации и минимальная возможность воздействия на результат, приоритетность в ранжировании уникальной информации, учет различных дополнительных критериев ранжирования.

При рассмотрении действующих поисковых систем таких как Rambler, Google, Yandex, были выявлены основные критерии ранжирования информации в них.

Так, Rambler, как первая система русскоязычного поиска, использует достаточно медленный, практически не изменившийся со времени основания алгоритм индексации страниц. При выдаче электронных документов заметен

приоритет к лексическому анализу уникальных больших по объему текстов, также учитываются мета тэги, огромное внимание уделяется свежести обновления информации на сайте, вес сайта оценивается в целом, как и в поисковой системе Yandex.

В поисковой системе Yandex, в последнее время особенно усилился критерий региональности, оптимизация всего сайта под отдельную ключевую фразу, так же увеличился приоритет уникальности контента. Выявлено что ссылки ведущие на страницы поднимают не саму страницу в поиске, а весь сайт в целом. Самый большой вес имеет по прежнему главная страница Интернет ресурса.

В ранжировании информации поисковой системой Google преобладают следующие значимые критерии: приоритет региональности, уникальности содержания, ссылки ведущие с «авторитетных ресурсов» на заданный сайт. Уникальность алгоритма Google в целом, является сочетание новых актуальных критериев, и технологий постраничного ранжирования PageRank, что определяет наиболее достоверный поиск. PageRank - это метод Google для измерения «важности» страницы. [2] Само название определяет алгоритм расчета цитируемости, разработанный и используемый by Sergey Brin и Larry Page, разработчиками поисковой системы Google.

Выявлено, что алгоритм ранжирования Google в корне отличается от алгоритмов Yandex, Rambler и является более достоверным, и стойким к воздействиям на результат внешними ссылками. Это можно заметить по ранжированию каждой страницы в отдельности, а не ресурса в целом, данный алгоритм требует гораздо большие вычислительные ресурсы, и имеет известную формулу опубликованную, его создателями. В выдаче также значительное внимание уделяется качеству содержания, введены критерии региональности выдачи, и разрабатываются новые методы, такие как формирования заголовков автоматически на базе контента страницы (снипет) [3], «уровень доверия» к сайту поисковой системой.

В результате анализа принципов и критериев ранжирования ведущих поисковых систем мы пришли к выводу, что поисковая система Google, объединяет новейшие критерии поиска информации и наиболее устойчивый к манипуляциям алгоритм ранжирования. При поиске информации система рассматривает каждую страницу Интернет, как значимую поисковую единицу информации.

Для представления механизмов работы алгоритма PageRank в замкнутой системе определения весов Интернет страниц разработана автоматизированная модель ранжирования электронных ресурсов, состоящая из сканирующей страницы паука, рассчитывающей PR составляющей и дополнительных критериев ранжирования. Определение эффективности влияния критериев ранжирования настраивается самим пользователем модели.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Принципы работы поисковой машины Рамблер [Электронный ресурс] / Gray // Энциклопедия поисковых систем – 2003. Режим доступа: <http://www.searchengines.ru/articles/004575.html>
- [2] Растолкованный PageRank [Электронный ресурс] / Chris Ridings, Jill Whalen, перевод Александра Садовского // Исследования и статистика – 2001. – Режим доступа: <http://www.digits.ru/articles/promotion/pagerank.html>
- [3] Search Snippets [Электронный ресурс] / Matt Cutt // SEM news - 2007. - Режим доступа: <http://blog.searchenginewatch.com/071127-105242>

## Задача анализа медицинской электрокардиограммы

*Лазейкин О. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Все методы для решения задач медицинской диагностики можно разделить на следующие два типа.

С одной стороны, разрабатываются системы, логика которых основана на статистических моделях — их основой служат математические алгоритмы, осуществляющие поиск обычно частичного соответствия между симптомами наблюдаемого пациента и симптомами наблюдавшихся ранее пациентов, диагнозы которых известны. Однако такие системы не имеют средств формирования понятного врачам объяснения полученного результата.

Системы второго типа основаны на знаниях экспертов. В них алгоритмы оперируют информацией о пациенте и знаниями о заболеваниях, представленных в некоторой форме. Именно такие системы способны сформулировать объяснение результата в медицинском контексте, то есть предоставить врачу результаты анализа данных, совокупность которых приводит к полученному решению задачи.

В работе рассматривается одна из задач медицинской диагностики, основанная на анализе электрокардиограммы.

Электрокардиограмма человека характеризуется набором временных и амплитудных параметров, в зависимости от которых ставится диагноз [1].

Основываясь на параметрах электрокардиограммы, требуется проанализировать классические методы постановки диагноза для наиболее распространенных заболеваний сердечно-сосудистой системы и программно реализовать один из них. Данная задача включает в себя два основных этапа: анализ данных и постановка диагноза на основе этих характеристик.

Регистрация происходящих в сердце биоэлектрических явлений дает характерную кривую — ЭКГ, которая схематически состоит из трех направленных вверх положительных зубцов P, R и T, двух отрицательных зубцов Q и S и непостоянного, направленного вверх зубца U. Кроме того, различают интервалы : P-Q, Q-T, S-T, T-P и два комплекса: QRS и QRST [1, 2].

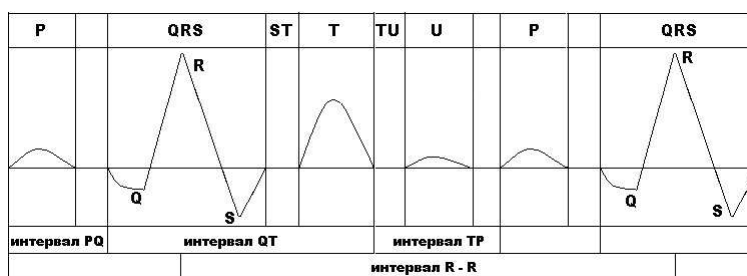


Рис. 1. Зубцы и интервалы нормальной ЭКГ

Обозначенные особенности (пики и интервалы) и являются стандартами описания электрокардиограммы человека.

В зависимости от положения электродов, с помощью которых производится съем электрокардиограмм, различают стандартные отведения — I, II, III, отведения от конечностей — aVR, aVL и aVF и одноплоскостные грудные отведения — V1, V2, V3, V4, V5, V6[1]. Для определения параметров ЭКГ обычно используют второе стандартное отведение. Под параметрами ЭКГ, как правило, понимают вычисленные временные и амплитудные значения пиков[2].

В медицинских источниках есть сведения о параметрах ЭКГ здорового человека. Обычно эти данные являются отправной точкой при анализе исследуемой электрокардиограммы. Предельные величины элементов ЭКГ в норме приведены в следующей таблице :

|     | Амплитуда зубцов, мм. |           |           |           |           |           |           |           |           |           |
|-----|-----------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
|     | $P_{min}$             | $P_{max}$ | $Q_{min}$ | $Q_{max}$ | $R_{min}$ | $R_{max}$ | $S_{min}$ | $S_{max}$ | $T_{min}$ | $T_{max}$ |
| I   | 0.3                   | 1.0       | 0.0       | 1.0       | 2.0       | 12.0      | 0.0       | 5.0       | 0.8       | 3.5       |
| II  | 0.9                   | 2.1       | 0.0       | 1.0       | 3.0       | 17.0      | 0.0       | 3.0       | 1.4       | 6.0       |
| III | -0.2                  | 1.0       | 0.0       | 2.0       | 0.5       | 15.0      | 0.0       | 4.0       | -1.4      | 4.9       |
| aVR | -1.5                  | -0.2      | 2.0       | 13.0      | 0.0       | 3.0       | 0.0       | 0.0       | -5.2      | -1.0      |
| aVL | 0.0                   | 0.9       | 0.0       | 2.5       | 0.5       | 9.0       | 0.0       | 9.0       | -0.5      | 2.0       |
| aVF | 0.5                   | 2.0       | 0.0       | 1.2       | 1.0       | 15.0      | 0.0       | 5.5       | 0.8       | 4.0       |
| V1  | -1.0                  | 0.5       | 0.0       | 0.0       | 0.5       | 5.2       | 3.0       | 15.0      | -3.8      | 2.0       |
| V2  | 0.0                   | 1.0       | 0.0       | 0.0       | 0.9       | 12.0      | 3.0       | 16.0      | -2.0      | 11.0      |
| V3  | 0.0                   | 1.0       | 0.0       | 0.0       | 1.1       | 19.1      | 0.0       | 16.0      | -2.0      | 13.3      |
| V4  | 0.2                   | 1.0       | 0.0       | 1.0       | 6.0       | 26.0      | 0.0       | 13.0      | 2.0       | 8.5       |
| V5  | 0.2                   | 1.0       | 0.0       | 1.0       | 5.0       | 24.7      | 0.0       | 7.0       | 1.8       | 6.8       |
| V6  | 0.2                   | 1.0       | 0.0       | 1.2       | 3.0       | 24.5      | 0.0       | 4.0       | 1.0       | 3.9       |

Таб. 1. Нормальная электрическая активность сердца

По временным и амплитудным характеристикам пиков и интервалов можно определить наличие тех или иных заболеваний у наблюдаемого пациента. Наиболее важную информацию несет пик R, в частности, именно по этому пику можно найти частоту сердечных сокращений[3].

## СПИСОК ЛИТЕРАТУРЫ

- [1] Тумановский М.Н., Бородулин Ю.Д., Никитин А.В., Фулки В.Б. Практическое руководство по электрокардиографии (с теоретическими основами) ч.1. –ВГУ, Воронеж, 1969.
- [2] Кечкер М.И. Электрокардиографические заключения и краткое описание изменений ЭКГ. –М.: Оверлей, 1993.
- [3] Сумароков А.В., Моисеев В.С., Михайлов А.А., Касымов И.Ю. Распознавание болезней сердца. –Т.: Медицина, 1978.

## Разработка плагина поиска лиц на фотоизображениях

*Максимова А. Н.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

В данной статье речь идет о создании плагина поиска лиц на растровых изображениях. Проблема распознавания лиц представляла повышенное внимание даже на самых начальных этапах развития систем распознавания образов. Не говоря уже о последних десятилетиях, за которые вместе с возрастающими возможностями компьютерной техники, расширился также и круг коммерческих и социальных задач, требующих применения идентификации человека по изображению его лица (системы контроля удостоверений личности, информационной безопасности, расследование криминальных событий, банковская сфера). Все это свидетельствует о неоспоримой актуальности данной проблемы.

Проблеме обнаружения лиц на изображениях посвящено множество работ ([4], [5], [6]). Существующие алгоритмы можно разделить на две основные категории – методы эмпирического распознавания и методы моделирования изображения лица. Методы первой категории основываются на опыте человека в распознавании лиц и пытаются его алгоритмизировать, для дальнейшего использования в построении автоматической системы распознавания. Методы второй категории основываются на теории распознавания образов, рассматривая задачу обнаружения лица как частный случай задачи распознавания с двумя предопределенными классами «лицо» и «не лицо». Большинство методов этой категории проходят в две стадии – стадию обучения и стадию классификации. Методы второй категории являются более эффективными, так как не нуждаются во вмешательстве человека на стадии обучения. За последние десять лет были предложены различные методы распознавания лиц, использующие такие подходы второй категории, как метод главных компонент (Shneiderman, Pentland), нейросети (Rowley, Kanade, Lyn), алгоритмы Viola-Jones (Viola, Lienhart, Whitehill), машины опорных векторов (Lienhart) и т.д. На текущий момент наиболее быстрым и точным алгоритмом в данной категории является алгоритм Viola-Jones[1], относящийся к семейству ассоциативных машин и способный классифицировать подаваемое изображение на два класса «лицо» и «не лицо». Стадия обучения требует предварительного обучения с учителем.

Основными отличиями данного алгоритма являются:



- представление изображения в виде «интегрального изображения»;
- построение классификатора основано на алгоритме AdaBoost [3];
- для распознавания используется каскад классификаторов.

Эффективность этого алгоритма подтверждена разработками многих авторов, что и послужило причиной его выбора для использования в нашем разрабатываемом плагине, на вход которого будет подаваться растровое изображение, а на выходе получать список прямоугольников, описывающих расположение лиц на изображении.

Неразрешенной осталась проблема обнаружения лица человека под произвольным углом поворота в режиме реального времени, которую алгоритм Viola-Jones не позволяет преодолеть.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Paul Viola, Michael J. Jones. Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137 – 154, 2004.
- [2] Вежневцев, А.Дегтярева «Обнаружение и локализация лица на изображении».
- [3] A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, no. 55. 1997.
- [4] H.A. Rowley, S. Baluja and T.Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell 20, pp 23-38., January 1998.
- [5] Jochen Maydt and Rainer Lienhart. Face detection with support vector machines and very large set of linear features, IEEE ICME 2002, Lousanne, Switzerland. pp xx-yy, Aug. 2002.
- [6] B.Moghaddam and A.Pentland, Probabilistic visual learning for object representation, IEEE Trans. Pattern Anal. Mach. Intell. 19(1), pp. 696-710, 1997.

## Эффективные методы решения переборных задач

*Мангилёв А. Ю.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

На данный момент не существует полиномиальных алгоритмов решения задач из класса NРС, однако такие задачи очень часто встречаются на практике. Любая подобная задача может быть решена полным перебором, имеющим экспоненциальную сложность. Обычно примитивный перебор не представляет трудности для реализации, однако время его работы на современных компьютерах слишком велико для решения большинства практических задач. Возникает необходимость оптимизации. Существует много способов повышения скорости работы программы, таких как:

- Отсечение ветвей дерева перебора, заведомо не приводящих к оптимальному результату.
- Перебор значительного числа случайных решений и проверка их на оптимальность.
- Применение различных эвристик.
- Использование динамического программирования с целью запоминания оптимального результата для подмножества.

Подробнее остановимся на методе динамического программирования, так как он позволяет асимптотически улучшить время работы. Применением других методов обычно сложно достичь заметного ускорения времени выполнения программы в худшем случае.

В каждой переборной задаче нужно найти подмножество некоторого множества, имеющее максимальную стоимость и удовлетворяющее определённому критерию. Прimitивный алгоритм перебирает все подмножества, оценивает их и выбирает оптимальное, что требует как минимум  $O(2^n)$  операций, где  $n$  - мощность множества  $A$ . Следующий метод позволяет получить оптимальное решение за время близкое к  $O(2^{n/2})$ .

Вне зависимости от конкретной задачи метод заключается в последовательном выполнении следующих этапов:

- (1) Разбиение множества  $A$  на 2 подмножества  $A_1$  и  $A_2$  приблизительно равной мощности.
- (2) Перебор всех подмножеств  $A_1$ ,  $A_2$  и сохранение результатов перебора. На этом шаге мы получаем 2 массива  $P_1$  и  $P_2$ , элементами которых являются пары  $\{m_i, c_i\}$ , где  $m_i$  — маска  $i$  — подмножества, а  $c_i$  — его оценка.
- (3) Поиск ответа на задачу. Оптимальное подмножество может целиком содержаться
  - во множестве  $A_1$ . Тогда оно уже найдено.
  - во множестве  $A_2$ . Тогда оно тоже уже найдено.
  - частично во множестве  $A_1$ , а частично в  $A_2$ . Для эффективной обработки этого случая необходимо:
    - (а) Упорядочить элементы массива  $P_2$  особым образом, различным для каждой конкретной задачи. Обычно это — сортировка по какому-то из параметров.
    - (б) Для каждого подмножества  $A$ , попавшего в  $P_1$ , используя упорядоченность элементов  $P_2$ , эффективным алгоритмом найти оптимальное дополняющее подмножество из  $P_2$ .

Рассмотрим метод на примере классической задачи о рюкзаке.

Задача формулируется следующим образом: дано множество  $A$  из  $n$  объектов и константа  $W$ . Каждый элемент  $a_i \in A$  характеризуется своей стоимостью  $p_i$  и весом  $w_i$ . Надо найти такое множество  $B = i_1, \dots, i_k$ , что  $\sum_k w_{i_k} \leq W$  и  $\sum_k p_{i_k} \rightarrow \max$ .

Разобьём множество на 2 подмножества  $A_1$  и  $A_2$  так что в первое попадёт  $n_1 = \lfloor n/2 \rfloor$  элементов, а во второе  $n_2 = \lceil n/2 \rceil$  элементов. Переберём все подмножества множеств  $A_1, A_2$  и сохраним все тройки  $\{m_i, w_i, p_i\}$ , где  $m_i, w_i, p_i$  - маска, вес и стоимость  $i$ -го подмножества, в массивах  $a$  и  $b$ . Пусть  $k = 2^{\lceil n/2 \rceil}$  число подмножеств  $A_2$ . Тогда массив  $b$  будет иметь размер  $k$ , а размер массива  $a$  будет отличаться от  $k$  не более чем в 2 раза. Проверим не оказалось ли оптимальное подмножество целиком в  $A_1$  или  $A_2$ , для чего найдём в массивах  $a$  и  $b$  максимальное  $p_i$ , такое что соответствующее  $w_i \leq W$ . Осталось проверить случай, когда оптимальное подмножество частично содержится в  $A_1$ , а частично в  $A_2$ . Для этого отсортируем все элементы из  $b$  по возрастанию  $w_i$ . Инициализируем массив  $m$ , для  $i = 0 \dots k - 1$ , следующим образом:  $m[i] = \max_{j=0 \dots i} b[j].p$ . Такую инициализацию можно осуществить за время  $O(k)$  следующим алгоритмом:

- 1:  $m[0] := b[0].p$
- 2: **for**  $i = 1, \dots, k - 1$  **do**
- 3:    $m[i] := \max(m[i - 1], b[i].p)$ ;

Теперь для каждой тройки  $\{m_i, w_i, p_i\}$  из  $a$  найдём бинарным поиском такое максимальное  $j$ , что  $b[j].w \leq d$ , где  $d = W - w_i$ , то есть вес который ещё можно добавить. К подмножеству  $m_i$ , без нарушения ограничения по суммарному весу, можно добавить все подмножества из  $b$  с индексами, не превосходящими  $j$  и только их. Но из всех таких подмножеств нас интересует только то, стоимость которого максимальна. А стоимость максимального уже записана в  $m[j]$  (вместе со стоимостью в  $m[j]$  можно хранить и само подмножество). Таким образом, зная  $j$  можно за константу определить оптимальное дополняющее подмножество.

Сложность этапов алгоритма:

- (1) Перебор всех подмножеств множеств  $A_1$  и  $A_2$ :  $O(2^{n/2})$  для каждого
- (2) Сортировка массива  $b$ :  $O(k * \log_2 k) = O(2^{n/2} * n)$
- (3) Построение массива максимумов:  $O(k) = O(2^{n/2})$
- (4) Бинарный поиск для каждого элемента из  $a$ :  $O(k * \log_2 k) = O(2^{n/2} * n)$ .

Итоговая оценка сложности алгоритма:  $O(2^{n/2} * n)$ , где  $n$  - мощность множества  $A$ , что существенно лучше оценки  $O(2^n)$ , которую имеет примитивный перебор. Данный алгоритм будет работать в  $2^{n/2}$  раз быстрее тривиального, что позволяет за приблизительно равное время решать задачу для  $n' = 2 * n$ .

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритмы: построение и анализ, 2-е издание. Издательский дом "Вильямс", 2007. — 386 с.
- [2] Дональд Э.Кнут. Искусство программирования. Издательский дом "Вильямс", 2006. — 365 с.

## Methods of Scene Management in Game Engine

*Pecek A.*

TAURIDA NATIONAL V. VERNADSKY UNIVERSITY  
FACULTY OF MATHEMATICS AND INFORMATICS  
CHAIR OF INFORMATICS (GROUP 401-I)

*This article deals with such notions of game engine architecture as scene management techniques. Definitions of methods and algorithms of 3d management and main benefits are given.*

#### INTRODUCTION

With the elaboration of computer graphics in the recent decades, there is persistent development of computer game industry. Computer games are created on the bases of architecture frameworks called game engines. Game engine represents a system of interconnected program components that is destined for realization and automatization all the main computer game processes such as media files loading, effect loading, rendering managing, resources distribution, etc. According with requests and restrictions by the hardware, the game engine system needs efficient control methods of virtual worlds' rendering. Therefore scene representation methods and algorithms of space partitioning were invented.

#### 1. HIERARCHICAL SCENE REPRESENTATION

For opportunity of scene managing and acceptable performance, there must exist a culling mechanism of invisible objects. The simplest checking algorithm of

3D object's attachment to camera's field of view is not efficient, since in this event checks for each 3D model's bounding volume (usually that is a bounding box) and the camera's view frustum intersections are needed.

A scene graph method allows to process the objects that grouped in hierarchies according with their spatial positions. The data structure represents a tree (but most popular name of this technology is remained the graph). The tree has nodes containing geometrical data and methods of provision the 3D model's bounding volume. The each node has only parent node and the set of arbitrary number of child nodes.

The scene organization according the scene graph method is important for computer games. There are short list of advantages of this approach below.

- Big parts of the scene may be culled. Since objects are connected with each other by spatial proximity principle then object in more high level can be represented as the structure union of its child objects. Therefore ablations of the huge unions are provided using fused bounding volumes.
- Opportunity of combined matrix world transformations using. The transformations go through from the parent to the child nodes.
- Selections the sub-scenes for shader effects application provided. For example, there is allowed to take light on the sampling group of objects that spatially connected. The effect is declared in parent node and then in all the children also, by recursive traversing.
- The scene organization by way of graph simplifies the game world saving and loading.

More the scene graph method described in [4].

## 2. SPACE MANAGEMENT

There exist algorithms of game world space partitioning that are performed mostly to resolve an only task - to cull as huge as possible scene parts. These methods give solution for reducing the number of objects for processing and rendering.

Depending of the game type and the gameplay mode, several basic methods of space partitioning are used.

- **Binary space partitioning (BSP)**. This method is based on creating a object-containing tree that is built from three-dimensional world. In order to this accomplish this method, the partitions are committed by some hyperplane (in 3D space this is a 2D plane). Space for root node is a pair of half-spaces divided by the plane contained in node. The first child node

called frontal contains the subset of world objects that situated on the positive side from this plane. And so the elements of second child node are found with the location on the negative side from the plane as the criterion. And so forth with partition building for each node in recursively traverse, the binary tree named BSP tree is built.

BSP tree can be used for several aims at one time. Firstly it allows to accomplish a space rejection of potentially invisible spaces. Secondly the BSP tree is used for collision detection of objects. The BSP method is expedient for use in indoor computer games (Quake, Unreal, Half-Life). More the method of BSP trees creating and ways of usage described in [1] and [2].

- **Quadrees and octrees.** These structure components are useful for simplification of object visibility checking process in the scene. The methods give a high-level mechanisms for representation the information about three-dimensional world and often are used jointly with the game's terrain system. The Quadrees and the octrees are applied for space partition into the cells. That approach allows to exclude the invisible scene components from processing.

The quadrees are built from planar divisions and the octrees are based on spacial ones. The Quadtree and the octree methods are mostly demanded in computer games with outdoor virtual worlds (Serious Sam, Crysis).

The quadtree method has been considered in [3] where is it's realization worked with the terrain processor.

- **Portals method.** This technology is applied in the games with room-structured implication i.e. there is a level structure, based on the availability of spaces, "rooms"and connecting them "door"openings. The meaning of method lies in real-time object lists creating that are "visible"from the camera point installed in some room and then adjustment the vision field by cut planes added according the door cutouts.

More this method described in [1].

### 3. DETAILS OF USING

The appliance of methods described above can be oversimplified with implementing them in the way of the object design and wide scalability architecture. This approach makes the scene graph method successfully interplaying with the space partition methods, since there is sufficient to

take space processing with sampling of visible cells as result and further take selection the scene subsets from the scene graph nodes which have the world coordinates reposed in the given visible cells.

If there is compound terrain used that consists of parts that are equal to the minimal quadtree cell, then the algorithm can provide the selected revealing of terrain's visible parts. This approach is very efficient since it ensures the high-level handling of computer game's details pruning.

The objects of computer game can supply wide animation actions, physics simulation and also AI algorithms. But without the methods of high-level processing there is a chance to lose all the occasions of very modern scene realization methods. The scene management algorithms is a good way to direct the visualization process and they ought to be applied in the first way.

#### CONCLUSION

The considered space management methods are widely demanded nowadays. They are used in most of successful computer game projects. Their importance is in reducing the number of processed virtual world's objects by the quite simply and quick ways.

#### REFERENCES

- [1] David H. Eberly. *3D Game Engine Design, Second Edition: A Practical Approach to Real-Time Computer Graphics*. – San Francisco: Morgan Kaufmann Publishers, 2006. – 1040 p.
- [2] Andre Lamothe. *Tricks of 3D Game Programming Gurus: Advanced 3D Graphics and Rasterisation*. – Indianapolis.: Sams Publishing, Indiana 46290, 2003. – 1728 p.
- [3] Greg Snook. *Real-Time 3D Terrain Engines using C++ and DirectX 9*. – Highham.: Charles River Media, 2003. – 374 p.
- [4] Garret Foster. *Understanding and implementing scene graphs*. // <http://www.gamedev.net/reference/programming/features/scenegraph/>

## Использование алгоритмов «Face Recognition» для индексирования и поиска фотоизображений

*Раунова Д. Р.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Интерес к процессам распознавания лиц в последнее время становится значительным в связи с возрастающими практическими потребностями: системы охраны, верификация кредитных карточек, криминалистическая экспертиза,

телеконференции, распознавание эмоций человека, создание систем машинного зрения в робототехнике и т.д.

Автоматическая идентификация личности по изображению на фотографии имеет широкое научное и коммерческое применение. Данная технология имеет особый интерес, так как с ее помощью распознавание может происходить бесконтактно в отличие от биометрических технологий распознавания по отпечаткам пальцев или по радужной оболочке глаза, и имеет преимущество перед системами идентификации личности с помощью паролей или документов (паспорта, водительских прав), которые слишком уязвимы и могут легко пострадать от подделки, воровства и других факторов [1].

Кроме того, в настоящее время в связи с доступностью фотоаппаратуры возникает необходимость создания систем, которые по фотоизображению человека будут находить все фотографии с ним на электронных носителях или в базах данных. *Целью работы является разработка программного комплекса, позволяющего осуществлять поиск фотографий по фотоизображению человека.*

На данный момент проблеме распознавания человека по изображению лица посвящено множество работ [1, 2, 3, 4, 5, 6], но эта задача еще полностью не разрешена. Основные трудности состоят в распознавании человека независимо от ракурса и условий освещенности при съемке, а также при различных изменениях, связанных с прической, возрастом и т.д. В связи с этим существует большое разнообразие алгоритмов и методов распознавания фотоизображений: нейросетевые методы, метод главных компонент, локальный дискриминантный анализ, фильтры Габора, сравнение эластичных графов, сравнение эталонов, анализ оптических потоков изображений, скрытые Марковские модели, а также методы, основанные на геометрических характеристиках лица.

Исторически первым подходом к решению проблемы автоматической идентификации человека по его фотоизображению был основан на выделении и сравнении некоторых антропометрических характеристик лица. Этот подход давно используется в криминалистике, однако замеры и сравнения выполняются обычно вручную.

Главная проблема, возникающая при решении задачи данным методом, — выбор совокупности характерных точек, наилучшим образом описывающих человеческое лицо, и способов надежного поиска таких точек [3]. При этом точки на лице, на которых основывается идентификация, не должны закрываться прической, бородой и т.п. Для обеспечения независимости процесса распознавания от масштаба изображения целесообразно описывать систему



идентификационных признаков в относительных единицах, выбранная система точек должна обеспечивать достаточную устойчивость процесса распознавания при незначительном изменении ракурса съемки и мимики. Кроме того, количество характерных точек системы должно быть минимальным.

В [4] была описана следующая система характерных точек и получаемых на их основе расстояний. Идентификационные точки: центры зрачков, кончик носа (центральная точка между носовыми отверстиями), центр рта (точка пересечения линии, разделяющей верхнюю и нижнюю губы объекта, и перпендикуляра, опущенного из точки, определяющей кончик носа объекта), левый угол левого глаза, правый угол правого глаза, нижние точки окончания мочек ушей. Расстояния между идентификационными точками, или «лицевые параметры»:

- расстояние между центром сетчатки правого глаза и точкой кончика носа;
- расстояние между центром сетчатки правого глаза и центром ротового отверстия;
- расстояние между центрами сетчатки глаз;
- расстояние между центром сетчатки левого глаза и точкой кончика носа;
- расстояние между центром сетчатки левого глаза и центром ротового отверстия;
- расстояние между центром ротового отверстия и кончиком носа.

Все выбранные параметры системы приняты в качестве основных, поскольку на них не влияют такие факторы, как прическа, макияж, наличие ювелирных украшений и пр. Используя шесть отобранных лицевых параметров можно составить до 30 линейных соотношений между ними, которые могут использоваться в качестве признаков. Основная причина использования параметров – независимость от масштаба и размера изображения. Однако самым главным достоинством геометрического подхода является возможность распознавания с его помощью человека по фотоизображениям, сделанным с большой разницей во времени [3]. В то же время основными трудностями по его применению являются автоматическая локализация антропометрических точек лица с требуемой точностью, оптимизация набора точек и вычисленных на его основе признаков. Возможным путем преодоления данных трудностей может быть использование фильтров Габора как для локализации определенных точек, так и для вычисления признаков, а также автоматическая оптимизация наборов полученных признаков.

## СПИСОК ЛИТЕРАТУРЫ

- [1] *Глазунов А.С.* Компьютерное распознавание человеческих лиц. //Зарубежная Радиоэлектроника. Успехи современной радиоэлектроники. Москва, 1997, № 8. –С. 3–14
- [2] *Кухарев Г.А.* Биометрические системы: Методы и средства идентификации личности человека. СПб.: Политехника, 2001. – 240с.
- [3] *Самаль Д.И.* Алгоритмы идентификации человека по фотопортрету на основе геометрических преобразований. Диссертация на соискание ученой степени кандидата технических наук. Минск, 2002. –С. 17–38.
- [4] *Самаль Д.И.* Построение систем идентификации личности на основе антропометрических точек лица //Сб. науч. тр. «Цифровая обработка изображений». Минск.: Ин-т техн. киберн. НАН Беларуси, 1999, –С. 105–114
- [5] *Bigun J., Chou K. and Olsson H.* Evidence on skill differences of women and men concerning face recognition //Lecture Notes in Computer Science. Vol.2091. Springer, 2001, –P. 44–51.
- [6] *Lawrence S., Giles C. L., Tsoi A. C. and Back A. D.* Face recognition: a convolutional neural network approach //IEEE Trans. on Neural Networks. –Special Issue on Neural Networks and Pattern Recognition, 1997, –P. 97–113.

## Методика преподавания программирования в учебных заведениях I-II уровней аккредитации

*Родько К. О.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Развитие новых технологий обучения, осуществление компьютеризации обучения обуславливают необходимость исследований, направленных на разработку учебных программ с применением ЭВМ и различных языков программирования и определение роли преподавателя в предмете. Учебный план ряда технических специальностей в учебных заведениях I-II уровней аккредитации предполагает последовательное изучение дисциплин «Основы информатики», «Программное обеспечение», «Программирование и алгоритмические языки» и другие.

Учебная программа дисциплины «Программирование и алгоритмические языки» определяет предмет изучения основ программирования с целью формирования у студентов теоретической базы в работе с алгоритмами и практических навыков создания программ. Полученные студентами знания и навыки работы с алгоритмическими языками должны служить базой для написания курсовой работы и изучения аспектов автоматизации работы системного администратора.

Рабочая программа дисциплины «Программирование и алгоритмические языки» содержит разделы:

- Алгоритмические языки;
- Язык программирования Object Pascal;
- Среда программирования Delphi;
- Алгоритмы поиска и сортировки.

Методика программирования в учебных заведениях I-II уровней аккредитации предполагает учет ряда факторов:

– важно разработать соотношение компьютерного и других видов обучения, соотношение практических работ, лекций и комбинированных занятий. Рабочая программа специальности «Обслуживание компьютерных систем и сетей» Симферопольского автотранспортного техникума рассчитана на 243 часа, из них 104 часа – лекции, 60 часов – практические работы, 79 часов на самостоятельное изучение;

– важно не забывать о характеристике и профилактике основных ошибок, допускаемых студентами при изучении программирования. Необходимо разработать систему проверки и контроля знаний по конкретной теме, например, проведение модульной контрольной работы или компьютерного тестирования, включающего теоретические и практические вопросы;

– чтобы воспитать необходимые профессиональные качества, нужно предоставить студентам возможность свободного творческого развития в рамках самостоятельно выполняемых проектов. Например, в нашей практике студенты могут проявить самостоятельность по теме «Сортировка массивов». Метод сортировки «пузырьком» подробно рассматривается на занятиях, а остальные методы даются студентам в качестве самостоятельного творческого задания.

Методика преподавания программирования должна постоянно меняться в зависимости от контингента обучаемых и стремительного развития данной отрасли. Каждый год приходят студенты с различным уровнем подготовки, и потому приходится менять подходы к объяснению некоторых тем, менять задания к практическим работам, чаще использовать прием «программирования по образцу».

Для стимуляции деятельности студентов важно объяснить им межпредметную связь, показать, что основы алгоритмического мышления будут им необходимы в дальнейшем обучении. Свои знания они смогут использовать при изучении в дальнейшем курса «Системного программирования», написании курсовой работы и прохождении практики по программированию.

В качестве изучаемого языка программирования на начальном уровне рассматривается Паскаль, который является эффективной основой для формирования алгоритмической культуры.

С методической точки зрения Паскаль хорошо подходит на роль учебного языка. Он позволяет ознакомиться с большинством понятий современного программирования и дает основы для изучения программирования в целом. Программы, написанные на Паскале, легко воспринимаются, что дает преимущество преподавателю при проверке практических работ. Еще одним преимуществом языка Паскаль является наличие системы анализа ошибок с подробными комментариями. На начальном этапе обучения программированию очень важно, чтобы студенты умели найти и исправить ошибки.

Классической схемой обучения является сначала изучение теоретических основ, а затем выполнение некоторых практических заданий. Занятия по программированию должны быть комбинированными, то есть включать в себя теоретическую часть, синтаксис, основные понятия и практическое заключение.

Чтобы научить студентов программированию, можно использовать «программирование по образцу». Сначала студентам предлагается готовая программа и объясняется ее цель и назначение, выполняется анализ команд и операторов. Затем студентам дается задание изменить существующую программу или написать похожую.

В практике преподавания нами используется стандартный подход, заключающийся в объяснении синтаксиса языка и основных понятий (необходимых теоретических сведений) и выполнении практического задания самостоятельно, а также «программирование по образцу» при объяснении наиболее сложных тем.

На занятиях по программированию следует добиваться того, чтобы все элементы программы были понятны студентам. Например, при объяснении темы «Алгоритмы сортировки массивов» нами используются диаграммы пошагового выполнения сортировки массива методом «пузырька».

В теоретической части занятия производится разбор алгоритма.

Исходный массив из 5 элементов заполняется случайными числами от 0 до 34.

|    |   |   |   |    |
|----|---|---|---|----|
| 14 | 4 | 9 | 1 | 33 |
|----|---|---|---|----|

На первом шаге сравниваются 1-й и 2-й элементы массива, и так как  $14 > 4$ , то они меняются местами, и массив принимает вид:

|   |    |   |   |    |
|---|----|---|---|----|
| 4 | 14 | 9 | 1 | 33 |
|---|----|---|---|----|

При этом объясняется необходимость вспомогательной ячейки для выполнения обмена. Затем сравниваются 2-ой и 3-ий элементы, так как  $14 > 9$ , то они тоже меняются местами:

|   |   |    |   |    |
|---|---|----|---|----|
| 4 | 9 | 14 | 1 | 33 |
|---|---|----|---|----|

И так далее. На последнем (четвертом) шаге сравниваются только 1-ый и 2-ой элементы:

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 4 | 9 | 14 | 33 |
|---|---|---|----|----|

На практических работах задания должны быть рассчитаны на студентов с разными уровнями подготовки. Например, задания к практической работе по теме «Сортировка массивов» разбиты на 3 варианта, каждый из которых содержит 5 заданий, и каждое правильно выполненное задание оценивается в один балл.

**Пример: Вариант №1**

**Задание 1:** Создать два одномерных массива А и В, состоящих из 10 элементов заполненных случайными числами в диапазоне от -25 до 25. Сформировать массив С такой что,  $C[i]=A[i]+B[i]$ .

**Задание 2:** Сформировать одномерный массив по следующему признаку: ячейки с чётными номерами заполняются числами, равными квадрату индекса, нечётные – нулями.

**Задание 3:** Составить программу, проверяющую, есть ли в массиве из 10 элементов две соседние ячейки, заполненные одинаковыми числами.

**Задание 4:** Составить программу, находящую максимальный элемент в массиве из 15 элементов, заполненном случайными числами в диапазоне от 0 до n, где n вводится с клавиатуры.

**Задание 5:** Составить программу, которая сортирует по убыванию массив из n элементов (n вводится с клавиатуры), заполненный случайными числами.

Имеющийся опыт преподавания программирования свидетельствует о необходимости постоянного совершенствования методики обучения, адаптации учебного материала к уровню знаний учащихся и использования различных дидактических приемов. Важным аспектом является проведение внеклассных мероприятий по программированию, таких как брейн-ринг и различные олимпиады с целью повышения качества знаний и развития у студентов интереса к предмету.

## СПИСОК ЛИТЕРАТУРЫ

- [1] *Андреева Е.В.* Методика обучения основам программирования на уроках информатики: Лекции 1-4, Москва Педагогический университет «Первое сентября», 2006. - 60 с.
- [2] *Андреева Е.В.* Методика обучения основам программирования на уроках информатики: Лекции 5-8, Москва Педагогический университет «Первое сентября», 2006. - 68 с.

## Технология создания шрифтов

*Скобелева Е. И.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Идея создания своего собственного шрифта или модификации уже существующего посещает каждого дизайнера. Актуальность создания новых шрифтов, в том числе и рукописных, подтверждается всё новыми и новыми шрифтами, создаваемыми по всему миру. На сегодняшний день существует множество фирм, которые предлагают свои услуги по созданию как рукописных шрифтов, так и «обычных» шрифтов, например для подчеркивания корпоративного стиля какого-нибудь предприятия, для персонификации электронных записей заказчика, для стилизованного оформления книжной или журнальной обложки. Создание шрифта – это длительный и трудоемкий процесс, требующий организованности и планирования. Дизайн шрифта можно отнести скорее не к художественным явлениям, а к явлениям, связанным с математикой и оптикой. Только построенный с учетом специфики человеческого зрения шрифт будет иметь оптимальные характеристики и наибольшую удобочитаемость. Без определенных вычислений здесь также не обойтись.

Целью работы является программная реализация процесса трассировки отсканированных символов, которая в дальнейшем может быть использована для создания шрифта в специализированных программах.

Реализация поставленной цели потребовала решения следующих задач:

- рассмотреть достоинства и недостатки векторной и растровой графики;
- изучить содержимое шрифтовых файлов;
- рассмотреть форматы хранения шрифтов;
- исследовать процесс создания нового шрифта при помощи специализированного программного обеспечения;
- изучить существующие шрифтовые технологии;

- реализовать процесс трассировки отсканированных символов.

Работа носит научно-исследовательский характер и состоит из трех основных разделов. В первом разделе приведен обзор существующей литературы по теме работы; рассмотрены достоинства и недостатки растровой и векторной графики; дано понятие компьютерного шрифта и изучены три основных типа компьютерных шрифтов: растровые, векторные и контурные. Рассмотрены основные форматы хранения шрифтов (PostScript, TrueType и OpenType) и проведен их сравнительный анализ. Исследован процесс создания нового шрифта при помощи специализированного программного обеспечения. Дана краткая характеристика основных программных пакетов, которые используются для преобразования растровой графики в векторную; приведены рекомендации по сканированию и правильной отрисовке контуров; рассмотрены программы, применяемые для генерации и редактирования шрифтов.

Второй раздел посвящен изучению и созданию новых шрифтовых технологий. В разделе рассмотрена задача контурного описания шрифта. Решение этой задачи составляет один из наиболее трудоемких этапов построения шрифта [1]. При автоматизации этого этапа предполагается, что полученные контуры являются черновыми и будут подвергнуты дальнейшей обработке рукой дизайнера, но при этом человек избавляется от рутинной трудоемкой операции по первичной подгонке кривых Безье к границе растрового изображения. Сделан вывод об автоматизации процесса хинтовки шрифта. Показано, что не следует рассчитывать на полностью автоматическое решение задачи. Подготовка высококачественного шрифта всегда будет требовать ручной доработки художника. Речь может идти лишь о повышении уровня автоматизации рутинных операций. Приведены примеры различных дефектов, которые могут быть найдены при непосредственной проверке качества хинтовки шрифта. Под качеством хинтовки шрифта понимается степень полноты и правильности инструкций по растеризации для всех начертаний, размеров и разрешающих способностей устройств отображения. Изучено моделирование рукописного шрифта с помощью жирных линий.

В третьем разделе приведен результат исследований, проведенных в рамках магистерской работы. Результатом является программная реализация, решающая задачу трассировки отсканированных символов.

Программная реализация включает в себя следующие этапы:

- 1) **Выделение символа** на изображении и преобразование бинарного раstra в матрицу, содержащую единицы на месте черных пикселей, и нули - на месте белых.

- 2) **Выделение контуров** символа с помощью алгоритма выделения контуров.
- 3) **Формирование цепочек.** На вход подается матрица символа, состоящая из нулей и единиц, на выходе образуется двумерный массив, представляющий собой цепочки пикселей, формирующих контур. Формирование цепочек происходит следующим образом: совершается обход контура по часовой стрелке, если встреченный черный пиксел имеет только одного черного соседа, то он заносится в текущую строку двумерного массива, и рассматривается его сосед. Если же пиксел «развилочный», то есть имеет нескольких соседей, он также заносится в текущую строку, после чего создается новая строка, в которую он заносится в качестве первого элемента, затем осуществляется переход на первого соседа (относительно обхода по часовой стрелке).
- 4) **Оптимизация полученного контура.** На этом этапе удаляются все лишние строки матрицы – содержащие единичные элементы и пары соседних точек (которые поглощаются более длинными отрезками, после чего удаляются).
- 5) **Аппроксимация полученных цепочек.** После этого этапа каждая из строк двумерного массива будет содержать только вершины графа, представляющего контур символа.

При помощи данной программы получено векторное представление символов, которое в дальнейшем может быть использовано в специализированных программах для генерирования шрифтов.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Местецкий Л.М.* Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. – М.: Физматлит, 2009. – 288 с.
- [2] Создание шрифта. – [http://wiki.auditory.ru/Создание\\_шрифта](http://wiki.auditory.ru/Создание_шрифта)



## Методы оценки информативных характеристик обучающей выборки

*Стародубов А. В.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. Вернадского  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

Понятие информативных признаков имеет большую роль в дискретных процедурах распознавания. Оценивая информативные признаки и выделяя наиболее представительные объекты, можно существенно повысить качество распознавания и снизить вычисляемые затраты.

Наиболее распространенной оценкой меры важности признака  $x_j, j \in \{1, \dots, n\}$ , рассматривается величина  $I_j = |C_j^A| \setminus |C^A|$ , где  $C_j^A$  – подмножество элементарных классификаторов из  $C^A$ , в которых содержится признак  $x_j$ ,  $A$  – тестовый алгоритм или алгоритм голосования по предварительным наборам. Однако такой способ оценки информативности признаков не всегда является хорошим, так как, если признак является тестом, то он входит только в один тупиковый тест и имеет малый вес согласно формуле для  $I_j$ , и в то же время указанный признак является важным и обладает большой информативностью, а малоинформативные шумящие признаки входят в большое число тестов, и как следствие имеют большой вес.

Пусть  $A$  – алгоритм голосования по предварительным наборам. Введя порог  $p_{min}$  минимальной встречаемости предварительного набора в базовой подвыборке для решения проблем шумящих признаков. Теперь будем рассматривать те представительные наборы, которые в базовой подвыборке встречаются не менее  $p_{min}$  раз, при этом шумящие многозначные признаки не получат большой оценки по информативности. Однако их информативный вес возможно будет довольно большим, поскольку не всегда можно взять большой порог  $p_{min}$ , так как структура класса такова, что каждый предварительный набор встречается небольшое число раз.

Для снижения вычислительной трудоемкости разделим случайным образом обучающую выборку на две подвыборки: базовую и контрольную. По базовой будем строить множество представительных наборов, а по контрольной – вычислять их веса. Вычислим веса предварительных наборов. Пусть  $\omega$  – предварительный набор класса  $K, K \in \{K_1, \dots, K_l\}$ , порождаемый парой  $(S', H)$ , где  $S'$  – объект из базовой выборки,  $\delta(K, \omega)$  – число объектов контрольной выборки, за которые представительный набор голосует “правильно“,  $\delta(\bar{K}, \omega)$  –

за которые голосует “неправильно“. В качестве функции, вычисляющей представительный набор, возьмем функцию

$$\nu(S', H) = \frac{1 + \delta(K, \omega)}{1 - \delta(\overline{K}, \omega)}.$$

Функция  $\nu(S', H)$  обладает следующими свойствами: монотонно возрастает по числу объектов из контрольной выборки, за которые представительный набор голосует “правильно“ и монотонно убывает по числу объектов, за которые голосует “неправильно“.

Для решения проблем шумящих признаков достаточно, чтобы оценка информативности шумящих признаков при разбиении (мощность контрольной и обучающей выборки постоянны, меняется только их состав) были неустойчивы по разбиению информации на две подвыборки, при том, что по остальным признакам наблюдается некоторая устойчивость.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *В.Е. Кузнецов*. Об одном стохастическом алгоритме вычисления информативных характеристик таблиц по методу тестов // Дискретный анализ. Новосибирск: ИМ СО АН СССР, 1973. Вып. 23. С. 8–23.
- [2] *Н.В. Песков*. Поиск информативных фрагментов описаний объектов в дискретных процедурах распознавания // Ж. вычисл. матем. и матем. физ. 2002. Том 42, С. 741–753.
- [3] *YU.I. Zhuravlev*. Diskrete methods of information analysis and algorithm synthesis // J.Pattern Recognition and Image Analys., 1997.V.7. P. 192–207.

## Построение цифровой модели местности по топографической карте

*Сушкович А. Б.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

В данной статье рассматривается один из возможных методов распознавания (векторизации) горизонталей на сканированных цветных топографических картах [1], а также приводится разработанный автором алгоритм быстрого построения трёхмерной модели соответствующей поверхности.

Рельеф местности изображается на топографической карте горизонталями (линиями уровня) – замкнутыми кривыми, проходящими через точки местности с одинаковой высотой над уровнем моря. Таким образом, чтобы получить трёхмерную модель данной местности, прежде всего нужно провести

автоматическое распознавание горизонталей, изображённых на карте. Обычно это означает наличие следующих этапов: (1) оцифровка; (2) бинаризация; (3) утончение, усечение, сглаживание; (4) собственно векторизация.

Рассмотрим каждый из приведённых этапов подробнее.

- (1) Оцифровка топографической карты путём сканирования. Желательно сканирование с как можно большим разрешением. Следует помнить, что разрешение растровой карты можно снизить после оцифровки, плохое же качество изображения способно катастрофически сказаться на результатах выполнения последующих шагов.
- (2) Бинаризация (отбор и порог, *filtering and thresholding*). Выделение чёрным цветом всех пикселей растровой карты, относящихся (скорее всего) к элементам рельефа, и белым – остальных (фона). Решение об отнесении точки к классу задающих рельеф принимается на основании её цвета (элементы рельефа обычно имеют коричневый цвет). Стоит отметить, что в результате выполнения данного шага некоторые линии уровня могут быть «разорваны»; вместе с тем, останутся многие элементы рельефа (обрывы, холмы, ямы), которые вовсе не будут учтены при построении трёхмерной модели средней детализации. Этот побочный эффект – плата за простоту критерия отбора.
- (3) Утончение (*thinning*), усечение (*pruning*), сглаживание (*smoothing*) изображения, распознавание текста. [2] В дальнейшем будет удобно, чтобы определяемые линии имели толщину не более одного пиксела. Для этого выполняется утончение. Усечение необходимо, поскольку линия шире одного пиксела, подвергнутая утончению, может приобрести ложные концевые точки. Сглаживание желательно, чтобы избежать чрезмерного ветвления диаграмм Вороного, которые будут строиться на этапе векторизации. Как утончение, так и усечение проводится с помощью морфологического преобразования «успех/неудача» по известным примитивам, а сглаживание – с помощью лапласиана. Кроме того, осуществляется автоматическое распознавание (а также последующее запоминание и удаление с карты) чисел, задающих высоты линий уровня. Условие, предъявляемое к методу: опознание чисел, расположенных под произвольным углом к границам карты.
- (4) Распознавание контурных линий. В результате выполнения описанных действий на растровой карте остаются (в виде чёрных пикселей) только линии, многие из которых, вероятно, незамкнутые. Необходимо объединить эти участки горизонталей в замкнутые контуры. Автору показался интересным подход, сущность которого заключается

в последовательном выполнении триангуляции Делоне объединённого множества точек горизонталей и вершин соответствующей диаграммы Вороного и удалении замкнувшихся контуров. [3] [1]

Результатом первого шага моделирования поверхности является множество её линий уровня, каждая из которых, в свою очередь, представляет собой пару объектов: множество соседних точек плоскости (на практике это означает соседство соответствующих пикселей) и уровень данной контурной линии. Такое представление поверхности всё ещё неудобно для её моделирования, поскольку неизвестны высоты точек, не принадлежащих горизонталям. Хотя на сегодняшний день существует несколько алгоритмов, позволяющих перейти от множества линий уровня к цифровой модели местности (digital terrain model), потребность в быстром построении модели, по-видимому, всё ещё есть. Автор предлагает простой, но эффективный (в смысле быстродействия) способ моделирования, состоящий в следующем: проводится триангуляция точек (возможно, не всех), составляющих контурные линии, затем каждый из полученных треугольников закрашивается с применением затенения (имеется в виду гладкий переход одного цвета в другой). Полученный растр называется картой высот; он представляет собой изображение, каждый пиксел которого имеет цвет, сопоставленный с конкретным значением высоты.

Основное преимущество данного подхода в возможности реализации встроенными средствами графической библиотеки OpenGL (в ходе внеэкрannого рендеринга), что гарантирует существенное снижение нагрузки на центральный процессор (за счёт использования графического акселератора); при этом переносимость программного обеспечения, реализующего данный алгоритм, по-прежнему остаётся высокой.

Триангуляция поверхности по карте высот, хотя и может иметь более или менее успешные реализации, в целом не представляет сложностей. Задача состоит в построении триангуляции карты высот (от выбора точек триангуляции зависит адекватность результирующей модели) и преобразовании её узлов в трёхмерные точки приданием им уровней (высот) в качестве аппликата (уровень точки отвечает цвету пиксела на карте высот).

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *Spinello Salvatore, Pascal Guittone. Contour Line Recognition From Scanned Topographic Maps // Journal of WSCG. – 2004. – Vol. 12, No. 1–3. – P. 419.*
- [2] *Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005. – 1072 с.*

- [3] *Nina Amenta, Marshall Bern and David Eppstein. The Crust and The  $\beta$ -Skeleton: Combinatorial Curve Reconstruction // Graphical Models and Image Processing. – 1998. – Vol. 60/2, No. 2. – Pp. 125–135.*

## Разработка параллельных приложений

*Тамазян А. А.*

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ им. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
КАФЕДРА ИНФОРМАТИКИ (ГРУППА 500-И)

В 1965 году, в процессе подготовки выступления Гордоном Муром (одним из основателей Intel), было высказано предположение, что число транзисторов на кристалле будет удваиваться каждые 24 месяца. Представив в виде графика рост производительности запоминающих микросхем, он обнаружил закономерность: новые модели микросхем разрабатывались спустя более или менее одинаковые периоды (18–24 мес.) после появления их предшественников, а ёмкость их при этом возрастала каждый раз примерно вдвое. Если такая тенденция продолжится, заключил Мур, то мощность вычислительных устройств экспоненциально возрастёт на протяжении относительно короткого промежутка времени.

И это действительно так. С тех пор вычислительные мощности процессоров неуклонно росли, и разработчики программного обеспечения стали к этому привыкать и на это рассчитывать. При написании приложения порой можно было пренебречь оптимизацией алгоритма, зная, что через некоторое время компьютеры все-равно станут мощнее. Это было правильно раньше, но потеряло свою актуальность сейчас. Ведь рано или поздно, разработчики процессоров должны были столкнуться с физическими ограничениями, такими как перегрев, потребление энергии и утечка электрического тока (при сокращении размеров транзисторов ее вероятность возрастает). Но это не значит, что прогресс остановился – он лишь пошел в другую сторону. До этого усилия были направлены на улучшение показателей в трех направлениях:

1. *Тактовая частота* – та же самая работа выполнялась быстрее с увеличением частоты;
2. *Оптимизация процесса выполнения* – как правило, выполнение большего количества операций за один цикл;
3. *Кэш* – увеличение размера кэш-памяти процессора, доступ к которой гораздо быстрее, чем к оперативной.

Улучшение в любом из пунктов автоматически положительно сказывалось на быстродействии последовательных (не параллельных, однопоточных) алгоритмов. Теперь же ситуация выглядит несколько иначе. Улучшения ведутся в следующих направлениях:

1. *Многопоточность* – это работа нескольких потоков параллельно на одном процессоре;
2. *Многоядерность* – это работа двух или более вычислительных ядер на одном процессорном кристалле.
3. *Кэш* – увеличение кеш-памяти будет продолжаться, по крайней мере, в ближайшем будущем.

Это значит, что на новых процессорах старые приложения будут работать не лучше, чем на старых (за исключением большего объема кэша). И, чтобы повысить скорость работы своих программ перед разработчиками становится задача распараллеливания используемых ими алгоритмов. Это влечет за собой ряд проблем.

Пример - потери производительности для организации параллелизма. Согласно *гипотезе Минского* (Minsky), ускорение, достигаемое при использовании параллельной системы, пропорционально двоичному логарифму от числа процессоров (т.е. при 1000 процессорах возможное ускорение оказывается равным 10). Безусловно, подобная оценка ускорения справедлива при распараллеливании определенных алгоритмов. Вместе с тем существует большое количество задач, при параллельном решении которых достигается практически 100%-е использование всех имеющихся процессоров параллельной вычислительной системы.

К тому же, на порядок увеличивается сложность отладки приложений, работающих в несколько потоков. Многие возникающие ошибки тяжело поддаются не только устранению, но и обнаружению.

Однако важность использования параллельных алгоритмов высока. Рассмотрим простой пример перемножения матриц A и B. Вот типичная последовательная реализация:

```
for (i=0; i<Size; i++)
  for (j=0; j<Size; j++)
    for (k=0; k<Size; k++)
      C[i*Size+j] += A[i*Size+k]*B[k*Size+j];
```

Для ее распараллеливания при помощи технологии OpenMP, достаточно добавить одну директиву *parallel for*:

```
#pragma omp parallel for private (j, k)
  for (i=0; i<Size; i++)
    ...
```

И мы уже получаем выигрыш в производительности примерно в 1,9 раза. Однако результат можно еще улучшить путем оптимизации работы с кэш-памятью, за счет разделения матриц на блоки. Блоки должны быть такого размера, чтобы в кэш-памяти одновременно помещалось целиком 3 блока (по одному от каждой матрицы). Благодаря этому все необходимые данные будут находиться в кэше:

```
int block = 250;
int grid = int (Size/double(BlockSize));
#pragma omp parallel for
for (int n=0; n<grid; n++)
  for (int m=0; m<grid; m++)
    for (int iter=0; iter<grid; iter++)
      for (int i=n*block; i<(n+1)*block; i++)
        for (int j=m*block; j<(m+1)*block; j++)
          for (int k=iter*block; k<(iter+1)*block; k++)
            C[i*Size+j] += A[i*Size+k] * B[k*Size+j];
```

Данная реализация увеличивает скорость операции перемножения матриц почти в 3,5 раза. На этом простом примере мы видим, как важна становится оптимизация и как незначительные усилия могут привести к значительному увеличению производительности.

Предстоит освоить новый подход к программированию. Подобно тому, как в 90х разработчики привыкали к принципам и особенностям ООП, сейчас настало время разобраться в тонкостях и идиомах параллельного программирования. Нужно научиться определять те части программы, которые можно распараллелить, делать это максимально эффективно, избегая при этом ошибок.

Проблематика параллельных вычислений является чрезвычайно широкой областью теоретических исследований и практически выполняемых работ и обычно подразделяется на следующие направления:

- **разработка параллельных вычислительных систем;**

- **анализ эффективности параллельных вычислений** для оценки *получаемого ускорения* вычислений и *степени использования* всех возможностей компьютерного оборудования при параллельных способах решения задач;
- **формирование общих принципов разработки параллельных алгоритмов** для решения сложных вычислительно трудоемких задач;
- **создание и развитие системного программного обеспечения для параллельных вычислительных систем**, которое позволяет разрабатывать параллельные программы и, кроме того, снизить в значительной степени остроту важной проблемы параллельного программирования – обеспечения мобильности (переносимости между разными вычислительными системами) создаваемого прикладного программного обеспечения;
- **создание и развитие параллельных алгоритмов** для решения прикладных задач в разных областях практических приложений.

Подводя итог, можно заключить, что параллельные вычисления являются перспективной (и очень привлекательной) областью применения вычислительной техники и представляют собой сложную научно-техническую область деятельности. Тем самым, знание современных тенденций развития аппаратных средств для достижения параллелизма, умение разрабатывать модели, методы и программы параллельного решения задач обработки данных следует отнести к числу важных квалификационных характеристик современного специалиста по информатике.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] *К.Ю.Богачев* Основы параллельного программирования. Москва: БИНОМ. Лаборатория знаний, 2003. – С. 42–75.
- [2] *В.П.Гергель*. Теория и практика параллельных вычислений. Москва: БИНОМ. Лаборатория знаний, 2007. – С. 3–25.
- [3] *Herb Sutter* The Free Lunch Is Over. Dr. Dobbs's Journal, 30(3), March 2005.
- [4] *Quinn M.J.* Parallel Programming in C with MPI and OpenMP. New York, NY: McGraw-Hill, 2004. – С. 5–30.



## СОДЕРЖАНИЕ

|   |    |
|---|----|
| Секция <b>МАТЕМАТИКИ</b> .....  | 3  |
| <b>Абибулаева С.А.</b> Компьютерные технологии в преподавании курса стереометрии в школе.....   | 3  |
| <b>Анисимова И.А.</b> Использование электронных учебников в преподавании курса планиметрии.....   | 5  |
| <b>Бабчинская Е.В.</b> Система $M/M/1$ с отключением сервера.....   | 6  |
| <b>Безруков М.Н.</b> Некоторые вопросы бесконечномерного анализа.....   | 10 |
| <b>Гажева И.А.</b> Полные линейные интегродифференциальные уравнения второго порядка в гильбертовом пространстве (параболический случай)..... | 11 |
| <b>Кандагура А.Н.</b> Горение вдоль полосы.....   | 14 |
| <b>Лапицкая Д.С.</b> Вольтеррово интегродифференциальное уравнение второго порядка (гиперболический случай).....                              | 18 |
| <b>Ли Е.В.</b> Исследование системы массового обслуживания с ненадежной линией.....   | 21 |
| <b>Литвиненко А.Л.</b> Статистические пакеты.....   | 24 |
| <b>Небибуллаева Э.А.</b> Исследование двухлинейной системы массового обслуживания с одним местом в очереди.....                               | 27 |
| <b>Негримовская А.Я.</b> Применение нейронных сетей для задач управления.....   | 29 |
| <b>Османова Э.А.</b> О некоторых свойствах ядер Фредгольма специального типа.....   | 34 |
| <b>Сёмкина Е.В.</b> Спектральные задачи, порождённые проблемами малых движений вязкоупругих и релаксирующих сред.....                         | 36 |
| <b>Сулейманова А.Р.</b> Анализ на временных шкалах.....   | 39 |

|  |    |
|--|----|
| <b>Хазова Ю.А.</b> Нелинейное интегральное уравнение Урысона первого рода .....  | 45 |
| <b>Шацкая И.С.</b> Спектральные свойства колейных операторов .....   | 47 |
| <b>Секция ПРИКЛАДНОЙ МАТЕМАТИКИ</b> .....  |    |
| <b>Абоимов С.А.</b> Применение технологии Swing для разработки кросс-платформенных приложений клиент-сервер .....                              | 51 |
| <b>Вышарев И.А.</b> Разработка объектных графических редакторов .....  | 53 |
| <b>Клевцова В.Н.</b> Моделирование гладких параметрических кривых и поверхностей с использованием интерполяции по Эрмиту .....                 | 56 |
| <b>Коцциенко А.В.</b> Защита информации в компьютерных сетях .....   | 60 |
| <b>Левинцов Н.Р.</b> Оптимизация архитектура и проектирование программных приложений в условиях жестких ограничений ресурсов .....             | 63 |
| <b>Лемтюжникова Д.В.</b> Разработка интеллектуализированной системы обработки информации по составным символам .....                           | 66 |
| <b>Литвиненко В.О.</b> Коррекционные программы для детей с нарушением зрения .....   | 70 |
| <b>Миланин А.А.</b> Разложение графа на трехсвязные компоненты .....   | 73 |
| <b>Намашко В.В.</b> Некоторые способы генерации поверхностей: разработка программного обеспечения .....  | 77 |
| <b>Соседко А.Ю.</b> Многокритериальная оптимизация архитектуры нейронных сетей, основанная на алгоритме обратного распространения ошибки ..... | 79 |
| <b>Секция ИНФОРМАТИКИ</b> .....  |    |
|  | 83 |

|   |     |
|---|-----|
| <b>Абдулхаиров А.Р.</b> Разработка поисковой системы на основе латентно-семантического индексирования .....       | 83  |
| <b>Буклерская Е.А.</b> Распознавание печатных и рукопечатных символов   | 85  |
| <b>Винц А.Б.</b> Дискретное извлечение корня .....  | 88  |
| <b>Dzhelialova A.I.</b> Monotonicity testing of boolean functions .....   | 91  |
| <b>Гусаров А.А.</b> Система Автоматического Манипулирования Сайтами   | 93  |
| <b>Koval A.</b> Some of statements and objectives of the agent-oriented modeling                                  | 95  |
| <b>Коринев М.В.</b> Анализ принципов и критериев ранжирования в современных поисковых системах .....              | 99  |
| <b>Лазейкин О.В.</b> Задача анализа медицинской электрокардиограммы   | 101 |
| <b>Максимова А.Н.</b> Разработка плагина поиска лиц на фотоизображениях .....                                     | 104 |
| <b>Мангилёв А.Ю.</b> Эффективные методы решения переборных задач .  | 105 |
| <b>Реуек А.</b> Methods of Scene Management in Game Engine .....  | 108 |
| <b>Раупова Д.Р.</b> Использование алгоритмов «Face Recognition» для индексирования и поиска фотоизображений ..... | 111 |
| <b>Родько К.О.</b> Методика преподавания программирования в учебных заведениях I-II уровней аккредитации .....    | 114 |
| <b>Скобелева Е.И.</b> Технология создания шрифтов .....   | 118 |
| <b>Стародубов А.В.</b> Методы оценки информативных характеристик обучающей выборки .....                          | 121 |
| <b>Сушкович А.Б.</b> Построение цифровой модели местности по топографической карте .....                          | 122 |
| <b>Тамазян А.А.</b> Разработка параллельных приложений .....  | 125 |