

ISSN 2222-8772

**М**АТЕМАТИЧЕСКИЕ  
**С**ТРУКТУРЫ  
И  
**М**ОДЕЛИРОВАНИЕ

№ 3(43)  
2017



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМ. Ф.М. ДОСТОЕВСКОГО»**

**МАТЕМАТИЧЕСКИЕ  
СТРУКТУРЫ  
И  
МОДЕЛИРОВАНИЕ**

**№ 3(43)**

Омск  
2017

**Математические структуры и моделирование.** — Омск : Омский государственный университет, 2017. — № 3(43). — 172 с.

ISSN 2222-8772 (print)

ISSN 2222-8799 (online)

#### **Редакционная коллегия**

|                           |                                                                                                                                                                                                                                              |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>А. К. Гуц</b>          | главный редактор, председатель редакционной коллегии, доктор физ.-мат. наук, профессор, зав. кафедрой кибернетики, Омский государственный университет им. Ф.М. Достоевского.                                                                 |
| <b>Д. Н. Лавров</b>       | ответственный за выпуск редактор, зам. глав. редактора, канд. техн. наук, доцент, зав. каф. компьютерных технологий и сетей, Омский государственный университет им. Ф.М. Достоевского.                                                       |
| <b>Н. Ф. Богаченко</b>    | технический редактор, зам. глав. редактора, канд. физ.-мат. наук, доцент, Омский государственный университет им. Ф.М. Достоевского.                                                                                                          |
| <b>С. В. Белим</b>        | д.ф.-м.н., профессор, зав. кафедрой информационной безопасности, проректор по научной работе, Омский государственный университет им. Ф.М. Достоевского.                                                                                      |
| <b>А. Ю. Веснин</b>       | д.ф.-м.н., профессор Новосибирского государственного университета, член-корреспондент РАН, заведующий лабораторией прикладного анализа, Институт математики им. С.Л. Соболева Сибирского отделения Российской академии наук, г. Новосибирск. |
| <b>В. П. Голубятников</b> | доктор физ.-мат. наук, профессор Новосибирского государственного университета, главный научный сотрудник Института математики СО РАН, г. Новосибирск.                                                                                        |
| <b>С. И. Горлов</b>       | доктор физ.-мат. наук, профессор, ректор Нижневартковского государственного университета.                                                                                                                                                    |
| <b>А. Г. Гринь</b>        | доктор физ.-мат. наук, профессор, кафедра кибернетики, Омский государственный университет им. Ф.М. Достоевского.                                                                                                                             |
| <b>В. А. Еровенко</b>     | доктор физ.-мат. наук, профессор, зав. кафедрой общей математики и информатики Белорусского государственного университета, г. Минск, Республика Беларусь.                                                                                    |
| <b>B. Zilber</b>          | Dr.Sc. (Phys.-Math.), Professor of Mathematical Logic, Mathematical Institute, University of Oxford, UK.                                                                                                                                     |
| <b>А. Н. Кабанов</b>      | канд. физ.-мат. наук, кафедра кибернетики, Омский государственный университет им. Ф.М. Достоевского.                                                                                                                                         |
| <b>П. А. Корчагин</b>     | доктор техн. наук, профессор, проректор по научной работе, Сибирская государственная автомобильно-дорожная академия (СибАДИ).                                                                                                                |
| <b>V. Kreinovich</b>      | Ph.D. (Phys.-Math.), Professor, Computer Science Department, University of Texas at El Paso, Texas, USA.                                                                                                                                     |
| <b>A. A. Fedorenko</b>    | Ph.D. (Phys.-Math.), Researcher (CR1) at the French National Centre of Scientific Research (CNRS) Laboratoire de Physique de l'ENS-Lyon, France.                                                                                             |
| <b>A. Jadczyk</b>         | Ph.D., Professor, Researcher, Laboratoire de Physique, Universite de Toulouse III et CNRS, France.                                                                                                                                           |

#### **Учредитель**

Федеральное государственное бюджетное  
образовательное учреждение  
высшего образования  
«Омский государственный университет  
им. Ф. М. Достоевского».  
Свидетельство о регистрации средства  
массовой информации ПИ № ФС77-57908  
от 28 апреля 2014 г. выдано Роскомнадзором.

#### **Адрес редакции, издателя и типографии**

644077, Россия, Омская обл., г. Омск,  
пр. Мира д. 55А.

**Дата выхода в свет:** 30.09.2017.

**Тираж** 100 экз.

Свободная цена.

## МАТЕМАТИЧЕСКИЕ СТРУКТУРЫ И МОДЕЛИРОВАНИЕ

Журнал основан в 1998 году. В журнале публикуются статьи, в которых излагаются результаты исследований по фундаментальной и прикладной математике, теоретической физике, компьютерным наукам, философии и истории математики и информатики, а также размышления, касающиеся окружающей нас природы и общества. Объекты исследования должны быть представлены в форме некоторых математических структур и моделей.

Все статьи журнала проходят обязательное рецензирование. Рефераты статей журнала опубликованы в «Реферативном журнале» и «Mathematical Reviews» (США). Журнал индексируется в РИНЦ (elibrary.ru) и «Zentralblatt für Mathematik» (Германия). Журнал входит в Перечень рецензируемых научных изданий ВАК РФ, в которых должны быть опубликованы основные результаты диссертаций на соискание учёных степеней (Приказ Минобрнауки России от 25 июля 2014 г. № 793).

Все статьи в журнале публикуются под лицензией Attribution 4.0 International (CC-BY).

Электронная версия журнала представлена в сети:

<http://msm.univer.omsk.su>  
<http://msm.omsu.ru>

Подписной индекс по каталогу «Пресса России»: 94082

Электронная почта главного редактора:

[guts@omsu.ru](mailto:guts@omsu.ru)

Электронная почта выпускающего редактора:

[lavrov@omsu.ru](mailto:lavrov@omsu.ru)

# СОДЕРЖАНИЕ

## Фундаментальная математика и физика

- А.К. Гуц. *Об источниках антигравитации* ... 5
- А.Н. Кабанов. *Центральный ряд группы унитарных автоморфизмов свободной алгебры Лейбница* ..... 12
- Т.К. Рустюмов, С.Т. Рустюмова. *Рёберная связность вершин полиэдров* ..... 16
- О. Kosheleva, V. Kreinovich. *Experimentally Observed Dark Matter Confinement Clarifies a Discrepancy in Estimating the Universe's Expansion Speed* ..... 29
- О. Kosheleva, V. Kreinovich. *Derivation of Gross-Pitaevskii Version of Nonlinear Schroedinger Equation from Scale Invariance* ..... 35
- А. М. Pownuk, V. Kreinovich. *Why Linear Interpolation?* ..... 43

## Прикладная математика и моделирование

- С.Н. Астраков, А.Г. Квашнин, Л.А. Короленко. *Построение эффективных сенсорных сетей с учётом стоимостных затрат* ..... 50
- Л.А. Володченкова, А.К. Гуц. *Равновесная динамика плодородия почвы в засушливых и влажных регионах* ..... 63
- Е.В. Ворожцов, В.П. Шапеев. *Применение интегральной формы уравнений коллокаций и дифференциальных условий согласования в методе коллокаций и наименьших квадратов* .. 70
- С.Н. Чуканов, С.В. Лейхтер. *Построение метаморфизмов растровых изображений на основе решения уравнений Эйлера-Пуанкаре* ... 86

## Компьютерные науки

- Е.А. Тюменцев. *О формализации процесса разработки программного обеспечения* ..... 96
- Ю.А. Осипова, Д.Н. Лавров. *Применение кластерного анализа методом  $k$ -средних для классификации текстов научной направленности* ..... 108

Продолжение на следующей странице



## Наши публикации



## Информационная безопасность

Д.М. Бречка, А.А. Литвиненко. *Исследование способов сокрытия информации в видеопотоке MPEG* ..... 122

А.К. Гуц, Е.П. Эннс. *Программа, моделирующая компьютерную сеть и сетевые атаки* .. 139

А.В. Костылев, Д.Н. Лавров, А.К. Гуц. *Идентификация суицидальных групп и нарушителей авторских прав в социальных сетях* .... 150

## ОБ ИСТОЧНИКАХ АНТИГРАВИТАЦИИ

А.К. Гуц

профессор, д.ф.-м.н., e-mail: guts@omsu.ru

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** Приводится формула для «гравитационной» силы в постоянном статичном сферически-симметричном пространстве-времени, позволяющая описывать смену гравитации на антигравитацию. Анализируются возможные источники антигравитации.

**Ключевые слова:** антигравитация, источник антигравитации.

Традиционно в общей теории относительности (ОТО) считается, что гравитация, т. е. притяжение материальных тел, определяется кривизной пространства-времени  $R_{ijkl} \neq 0$ . В силу этого компоненты метрического тензора  $g_{ik}$  рассматриваются как потенциалы гравитационного поля. Однако, как учит диалектика, притяжения не бывает без отталкивания. Следовательно, кривизна пространства-времени должна, в зависимости от ситуации, проявляться то как гравитация, то как антигравитация.

Следует иметь в виду, что поскольку исторически ОТО создавалась как теория релятивистской гравитации, т. е. как теория притяжения, то отталкивание, т. е. антигравитацию в ОТО часто называют гравитационным отталкиванием, а собственно гравитацию — гравитационным притяжением. Автору кажется уместным использовать термины гравитация и антигравитация.

В статье мы выводим формулу, показывающую, каким образом кривизна сказывается либо как притяжение, либо как отталкивание.

### 1. Антигравитация определяется знаком кривизны Риччи

#### 1.1. Две формулы для постоянного метрического поля

Для постоянного метрического, или «гравитационного», поля  $g_{ik}$  ( $g_{00} > 0$ ) справедливы следующие две формулы:

$$f_{\alpha} = \frac{mc^2}{\sqrt{1 - \frac{V^2}{c^2}}} \left\{ -\frac{\partial}{\partial x^{\alpha}} \ln \sqrt{g_{00}} + \sqrt{g_{00}} \left[ \frac{\partial}{\partial x^{\beta}} \left( \frac{g_{0\alpha}}{g_{00}} \right) - \frac{\partial}{\partial x^{\alpha}} \left( \frac{g_{0\beta}}{g_{00}} \right) \right] \frac{V^{\beta}}{c} \right\} \quad (1)$$

— для силы поля  $g_{ik}$ , действующей на пробную частицу [1, р.327], где  $V^{\alpha}$  — скорость частицы, и

$$\frac{1}{g_{00}} R_{00} = \frac{1}{\sqrt{g_{00}}} (\sqrt{g_{00}})_{;\alpha}^{\alpha} + \frac{\sqrt{g_{00}}}{4} f_{\alpha\beta} f^{\alpha\beta} \quad (2)$$

— для 00-компоненты уравнений Эйнштейна [1, с. 361], где

$$f_{\alpha\beta} = \frac{\partial}{\partial x^\alpha} \left( \frac{g_{0\beta}}{g_{00}} \right) - \frac{\partial}{\partial x^\beta} \left( \frac{g_{0\alpha}}{g_{00}} \right)$$

и «;» обозначает ковариантную или контрвариантную производную относительно 3-метрики [1]

$$\gamma_{\alpha\beta} = -g_{\alpha\beta} + \frac{g_{0\alpha}g_{0\beta}}{g_{00}}$$

и соответствующей римановой связности

$$\lambda_{\beta\nu}^\alpha = \frac{1}{2} \gamma^{\alpha\mu} \left( \frac{\partial \gamma_{\beta\mu}}{\partial x^\nu} + \frac{\partial \gamma_{\nu\mu}}{\partial x^\beta} - \frac{\partial \gamma_{\beta\nu}}{\partial x^\mu} \right).$$

## 1.2. Статичное метрическое поле

Рассмотрим статичное метрическое поле  $g_{ik}$ , т. е. компоненты  $g_{ik}$  независимы от времени  $x^0$  и  $g_{0\alpha} = 0$  ( $\alpha = 1, 2, 3$ ).

Полагаем также, что поле  $g_{ik}$  описывается уравнениями Эйнштейна вида

$$R_{ik} - \frac{1}{2} g_{ik} R = \frac{8\pi G}{c^4} [(\varepsilon + p)u_i u_k - p g_{ik}],$$

$$u^0 = \frac{1}{\sqrt{g_{00}} \sqrt{1 - \frac{v^2}{c^2}}} - \frac{g_{0\alpha} v^\alpha}{c \sqrt{1 - \frac{v^2}{c^2}}}, \quad u^\alpha = \frac{v^\alpha}{c \sqrt{1 - \frac{v^2}{c^2}}}, \quad (3)$$

а материя его создающая покоится в нашей системе отсчёта, т. е.  $v^\alpha = 0$  ( $\alpha = 1, 2, 3$ ).

Тогда формулы (1), (2) принимают вид:

$$f_\alpha = -\frac{mc^2}{\sqrt{1 - \frac{v^2}{c^2}}} \frac{\partial}{\partial x^\alpha} \ln \sqrt{g_{00}} = -\frac{mc^2}{\sqrt{g_{00}} \sqrt{1 - \frac{v^2}{c^2}}} \frac{\partial \sqrt{g_{00}}}{\partial x^\alpha}, \quad (4)$$

$$\frac{1}{g_{00}} R_{00} = \frac{1}{\sqrt{g_{00}}} (\sqrt{g_{00}})_{;\alpha}^{\alpha}. \quad (5)$$

Из (4) и (5) выводим:

$$\begin{aligned} \frac{1}{g_{00}} R_{00} &= \frac{1}{\sqrt{g_{00}}} (\sqrt{g_{00}})_{;\alpha}^{\alpha} = \frac{1}{\sqrt{g_{00}}} \left( \frac{\partial \sqrt{g_{00}}}{\partial x^\alpha} \right)^{;\alpha} = -\frac{\sqrt{1 - \frac{v^2}{c^2}}}{mc^2 \sqrt{g_{00}}} (\sqrt{g_{00}} f_\alpha)^{;\alpha} = \\ &= -\frac{\sqrt{1 - \frac{v^2}{c^2}}}{mc^2 \sqrt{g_{00}}} \gamma^{\alpha\beta} (\sqrt{g_{00}} f_\alpha)_{;\beta} = -\frac{\sqrt{1 - \frac{v^2}{c^2}}}{mc^2 \sqrt{g_{00}}} (\sqrt{g_{00}} \gamma^{\alpha\beta} f_\alpha)_{;\beta} = -\frac{\sqrt{1 - \frac{v^2}{c^2}}}{mc^2 \sqrt{g_{00}}} (\sqrt{g_{00}} f^\beta)_{;\beta}. \end{aligned}$$

Используем формулу римановой геометрии

$$A_{;\beta}^\beta = \frac{1}{\sqrt{\gamma}} \frac{\partial (\sqrt{\gamma} A^\beta)}{\partial x^\beta}, \quad \gamma = \det ||\gamma_{\alpha\beta}||.$$

Значит,

$$\frac{1}{g_{00}} R_{00} = -\frac{\sqrt{1 - \frac{V^2}{c^2}}}{mc^2 \sqrt{g_{00}} \sqrt{\gamma}} \frac{\partial(\sqrt{g_{00}} \sqrt{\gamma} f^\beta)}{\partial x^\beta}.$$

Кривизна Риччи в направлении  $u^i$  определяется как  $Ric(u) = R_{ik} u^i u^k$ .

В нашем случае  $u^i = (u^0, 0, 0, 0)$ ,  $u^0 = 1/\sqrt{g_{00}}$  (см. (3)) и

$$Ric(u) = R_{00} u^0 u^0 = -\frac{\sqrt{1 - \frac{V^2}{c^2}}}{mc^2 \sqrt{g_{00}} \sqrt{\gamma}} \frac{\partial(\sqrt{g_{00}} \sqrt{\gamma} f^\beta)}{\partial x^\beta}, \quad (6)$$

или

$$\frac{\partial(\sqrt{-g} f^\beta)}{\partial x^\beta} = -\frac{mc^2}{\sqrt{1 - \frac{V^2}{c^2}}} \sqrt{-g} Ric(u), \quad (7)$$

поскольку  $g_{00}\gamma = -g$ .

### 1.3. Статичное сферически симметричное метрическое поле

Для этого метрического поля  $f^\alpha = (f^r, f^\theta, f^\varphi)$ ,  $f^\theta = f^\varphi = 0$ . Тогда из (7) имеем

$$\frac{d}{dr} \sqrt{-g} f^r = -\frac{mc^2}{\sqrt{1 - \frac{V^2}{c^2}}} \sqrt{-g} Ric(u),$$

или

$$f^r = -\frac{mc^2}{\sqrt{1 - \frac{V^2}{c^2}}} \int_0^r \sqrt{-g} Ric(u) dr. \quad (8)$$

с учётом того, что  $f^\alpha(0) = 0$ .

Формула (8) — это искомая нами формула. Она показывает, каким образом кривизна пространства-времени сказывается как сила, действующая на пробные тела.

Мы видим, что изменение знака кривизны Риччи в определённой области пространства-времени превращает гравитационное притяжение в гравитационное отталкивание, и наоборот.

О том, что знак кривизны Риччи определяет свойство метрики (гравитации)  $g_{ik}$ , проявляется как притяжение или отталкивание в смысле, что соседние геодезические вблизи любой точки приближаются в среднем друг к другу или отдаляются друг от друга, известно давно [2, р. 531], и в случае констатации притяжения условие положительности кривизны Риччи называется энергетическим условием. В нашей работе речь идёт о притяжении или отталкивании пробного тела от самого источника гравитационного поля, скажем Земли.

#### 1.4. Формула Грёна

Если принять во внимание формулу [1, р. 361] в форме

$$\frac{1}{g_{00}} R_{00} = \frac{8\pi G}{c^4} \left( \frac{\varepsilon + p}{1 - \frac{v^2}{c^2}} - \frac{\varepsilon - p}{2} \right), \quad (9)$$

то для покоящейся материи

$$Ric(u) = \frac{4\pi G}{c^4} \cdot (\varepsilon + 3p),$$

и формула (8), может быть записана как

$$f^r = - \frac{4\pi m G}{c^2 \sqrt{-g} \sqrt{1 - \frac{v^2}{c^2}}} \int_0^r \sqrt{-g} (\varepsilon + 3p) dr, \quad (10)$$

Эта формула обобщает формулу Грёна [3, формула (9)], а интеграл в правой части, описывающий «активную гравитационную массу», часто именуется формулой Толмена-Уиттекера [3, 4].

Формула (10) говорит, что при изменении знака величины  $(\varepsilon + 3p)$  в определённой области  $V$  пространства притяжение (гравитацию) меняется на отталкивание (антигравитация), и обратно. Об этом писал Грэн в 1985 году. Он не называл источник отталкивания, хотя указал на интересную статью Коэнов [5], где гравитационное отталкивание возникает в электрически заряженных оболочках.

Из формулы (10) видно, что источником отталкивания является либо появление в области  $V$  экзотической материи ( $\varepsilon < 0, \varepsilon + 3p < 0$ ), либо большого отрицательного давления ( $\varepsilon > 0, p < 0, \varepsilon + 3p < 0$ ).

Заметим, что формула (10) отражает физическую сторону описания движения пробных тел, а формула (8) — геометрическую. Связь же геометрии и физики осуществляется посредством уравнений Эйнштейна.

## 2. Материя, обеспечивающая смену знака кривизны Риччи

Формула (8) позволяет утверждать, что универсальная сила гравитации (притяжения) наблюдается в пространственно-временных областях, иначе говоря (с точки зрения формулы (10)), в материальных средах, для которых кривизна Риччи положительна. В средах, где кривизна Риччи отрицательна, на пробные тела действует сила антигравитации (отталкивания). Мы живём в материальной среде, где властвует гравитация. Естественно поискать среды, где универсальной силой воздействия на тела будет антигравитация.

## 2.1. Экзотическая материя

Об экзотической материи, т. е. о материи с отрицательной плотностью писал ещё Эйнштейн [6, с. 89]. Отрицательная масса — это гипотетическая материя, масса которой имеет противоположное значение массе нормального вещества. Такое вещество, если бы оно существовало, нарушало бы одно или несколько энергетических условий, в частности, меняя знак кривизны Риччи и, следовательно, проявляя эффект антигравитации и некоторые другие странные свойства. Считается, что вещество с отрицательной массой можно использовать для создания 3-мерных кротовых нор в пространстве.

Недавно было объявлено от создании жидкости (конденсата Бозе-Эйнштейна в объёме менее  $0,001 \text{ мм}^3$ ), частицы которой приобретали отрицательную массу [7].

## 2.2. Эффект Казимира

Наиболее известным представителем экзотической материи является вакуум в области с отрицательным давлением, производимыми эффектом Казимира [8, с. 393-394],

$$\varepsilon = -\frac{\pi}{24a^2}, \quad \varepsilon = p.$$

Эффект Казимира, проявляющийся в притягивании двух параллельных пластин в вакууме, можно описать как наличие «отрицательного давления» между пластинами, когда вакуум лишён не только обычных, но и части виртуальных частиц (их рождение подавляется), т. е. «откачали всё и ещё чуть-чуть»<sup>1</sup>. На расстояниях порядка 10 нм — сотни размеров типичного атома — давление, создаваемое эффектом Казимира, оказывается сравнимым с атмосферным. Сила Казимира растёт как  $a^{-4}$  при уменьшении расстояния  $a$  между пластинами. Соответственно растёт сила антигравитации (как  $a^{-2}$ ).

Возможно, из пластин можно сконструировать нечто похожее на полостную структуру, описанную В.С. Гребенниковым, которая обладала антигравитационными свойствами, если верить автору [9].

Силу Казимира без традиционного квантового её описания изучил Е.М. Лифшиц. Он рассмотрел заполненное ван-дер-ваальсовой жидкостью пространство между пластинами и показал, что при определённых  $\varepsilon_1, \varepsilon_2$  диэлектрических проницаемостях пластин и жидкости  $\varepsilon_3$  возникает не притяжение пластин, а отталкивание. Эту силу называют силой Казимира-Лифшица. Она была измерена (жидкость — бромбензол) в 2008 и в 2012 годах [10, 11]. Авторы эксперимента писали о том, что отталкивающая сила Казимира-Лифшица позволяет осуществлять квантовую левитацию объектов в жидкости.

Обратим внимание на то, что эффект Казимира даёт в наше распоряжение среду не только с экзотической материей, но и с отрицательным давлением. Часто говорят о том, что эта среда есть среда с тёмной энергией, которой

---

<sup>1</sup>Эффект Казимира. — Викиверситет.

свойственно гравитационное отталкивание, т. е. это среда, где тела не притягиваются, а отталкиваются.

### 2.3. Отрицательное давление и гравитационное отталкивание

Э.Б. Глинер [12] в 1965 году ввёл понятие вакуумоподобного состояния вещества с отрицательным давлением  $p$ , положительной плотностью энергии  $\varepsilon$  с уравнением состояния  $\varepsilon = -p$ . Под отрицательным давлением он понимал внутренние объёмные силы в материи, определяющие не силы отталкивания (как и для среды, доступной для наблюдения, состоящей из частиц), а силы притяжения.

Отрицательное давление считается атрибутом тёмной энергии, понятие о которой появилось в космологии в конце XX века. Оно приводит к появлению отталкивающих гравитационных сил, о которых «неспециалисты иногда говорят как об антигравитации» [13]. Лучше говорить о космологической антигравитации, обеспечивающей инфляцию Вселенной, т. е. её ускоренное расширение вопреки (притягивающим) силам гравитации.

Как давление может быть отрицательным?

Давление обычного вещества, как известно, связано с движением молекул. Ударяясь о стенку сосуда, молекулы газа передают ей свой импульс, отталкивают её, давят на неё. Газ стремится расширить объём своего пребывания.

В среде с отрицательным давлением всё происходит иначе. Аналогией отрицательного давления и порождаемой им антигравитации служит вертикально натянутая резиновая плёнка, зажата снизу в тиски. Каждый её квадратный сантиметр растянут и стремится сжаться. Плёнка имеет натяжение  $p_1 < 0$ . Пометим на ней точку  $A$ , находящуюся на расстоянии  $l$  от тисков. Потянем плёнку сильнее вверх. Натяжение, т. е. отрицательное давление  $p_2$ , по модулю усилится до  $|p_2| > |p_1|$ . Точка  $A$  удалится от тисков ещё дальше вверх, как будто под воздействием силы  $f$ , поднявшей точку выше над тисками. Это и есть иллюстрация действия силы антигравитации, поднимающей материальные точки все выше вверх и являющейся **естественной силовой характеристикой воздействия среды с отрицательным давлением на материальные точки**.

Сходным образом ведут себя галактики во Вселенной под действием отрицательного давления тёмной энергии. Вселенная не сжимается, а ускоренно расширяется. Получается, что материя, среда с очень сильным отрицательным давлением, парадоксальным образом не сжимается, а наоборот, распухает под действием, и это следствие влияния отрицательности кривизны Риччи.

Таким образом, можно ответить на вопрос: почему сильнейшее отрицательное давление вместо сжатия приводит к расширению, или почему вместо падения вниз наблюдается падение вверх? Таковы свойства кривизны пространства-времени, выраженные уравнением Эйнштейна.

## ЛИТЕРАТУРА

1. Ландау Л.Д., Лифшиц Е.М. Теория поля. М., 1967.
2. Hawking S.W., Penrose R. The Singularities of Gravitational Collapse // Proc. Royal Soc. Lond. A. 1970. Vol. 314. P. 529–548.
3. Grøn Ø. Repulsive gravitation and electron models // Phys. Rev. D. 1985. V. 31, No. 8. P. 2129–2131.
4. Tolman R.C. On the Use of the Energy-Momentum Principle in General Relativity // Phys. Rev. 1930. V. 35, Iss. 8. P. 875–895.
5. Cohen J.M., Cohen M.D. Exact Fields of Charge and Mass Distributions in General Relativity // Nuovo Cimento. 1969. V. 60. P. 241–248.
6. Эйнштейн А. Собрание научных трудов. Т. 2. М. : Наука, 1966.
7. Khamsehchi M.A., Hossain Kh., Mossman M.E., Zhang Y., Busch Th., McNeil Forbes M., Engels P. Negative-Mass Hydrodynamics in a Spin-Orbit–Coupled Bose-Einstein Condensate // Phys. Rev. Lett. 2017. V. 118. P. 155301; arXiv:1612.04055.
8. Мостепаненко В.М., Трунов Н.Я. Эффект Казимира и его приложения // УФН. 1988. Т. 156, вып. 3. С. 385–426.
9. Гребенников В.С. Мой мир. Новосибирск : Советская Сибирь, 1997.
10. Munday J.N., Capasso F., Parsegian V.A. Measured long-range repulsive Casimir–Lifshitz forces // Nature. 2009. No. 457. P. 170–173.
11. Bostrom M., Sernelius Bo E., Baldissera G., Persson C., Ninham B.W. Casimir–Lifshitz interaction between  $ZnO$  and  $SiO_2$  nanorods in bromobenzene turns repulsive at intermediate separations due to retardation effects // Phys. Rev. 2012. A85. P. 044702. arXiv.org (arXiv:1612.04055).
12. Глинер Э.Б. Алгебраические свойства тензора энергии-импульса и вакуумоподобные состояния вещества // ЖЭТФ. 1965. Т. 49(8). С. 542–548.
13. Лукаш В., Михеева Е. Тёмная энергия Вселенной // Вокруг света. 2008. № 9.

## ON SOURCES OF ANTIGRAVITATION

**A.K. Guts**

Dr.Sc. (Phys.-Math.), Professor, e-mail: guts@omsu.ru

Dostoevsky Omsk State University

**Abstract.** The formula for the “gravitational” force in a constant static spherical-symmetric space-time, allowing to describe the change of gravity to antigravity, is given. Possible sources of antigravitation are analyzed.

**Keywords:** antigravitation, source of antigravitation.

*Дата поступления в редакцию: 23.08.2017*



## ЦЕНТРАЛЬНЫЙ РЯД ГРУППЫ УНИТРЕУГОЛЬНЫХ АВТОМОРФИЗМОВ СВОБОДНОЙ АЛГЕБРЫ ЛЕЙБНИЦА

А.Н. Кабанов

к.ф.-м.н., e-mail: m01kab@mail.ru

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** Получено описание центрального ряда группы унитарных автоморфизмов свободной алгебры Лейбница над произвольным полем.

**Ключевые слова:** алгебра Лейбница, унитарный автоморфизм, гиперцентр.

В статье автора [1] описывалось строение гиперцентральной серии подгруппы унитарных автоморфизмов, выделяемой в группе всех автоморфизмов свободной метабелевой алгебры Ли. В работах автора [2, 3] описывалось строение гиперцентральной серии аналогичной подгруппы свободной алгебры Ли.

Ж.Л. Лоде ввёл понятие алгебры Лейбница [4] как обобщение понятия алгебры Ли. Теория алгебр Лейбница активно развивается, и многие результаты из теории алгебр Ли переносятся на алгебры Лейбница.

В данной работе представлено частичное описание центральной серии группы унитарных автоморфизмов свободной алгебры Лейбница.

Напомним, что неассоциативная алгебра  $L$  над полем  $F$  с билинейным произведением  $[\cdot, \cdot]$  называется (правой) алгеброй Лейбница, если для любых элементов  $x, y, z \in L$  выполняется (правое) тождество Лейбница:

$$[[x, y], z] = [x, [y, z]] + [[x, z], y].$$

Или, что то же самое,

$$[x, [y, z]] = [[x, y], z] - [[x, z], y].$$

Отсюда видно, что  $[x, [y, y]] = 0$ .

Кроме того, видно, что введение свойства антикоммутативности превращает тождество Лейбница в тождество Якоби, а алгебру Лейбница — в алгебру Ли. Поэтому алгебру Лейбница часто называют «некоммутативным» обобщением алгебры Ли.

Из тождества Лейбница также следует, что любой элемент алгебры  $L$  можно представить как линейную комбинацию элементов вида  $[[[a, b], c], d], \dots$ , поэтому для краткой записи будем опускать скобки, положив

$$[[a, b], c] = abc.$$

Более того, примем записи

$$[[a, b], b] = ab^2, [[[a, b], b], b] = ab^3 \text{ и т. п.}$$

Пусть  $L_n$  — свободная алгебра Лейбница над полем  $F$  с множеством свободных порождающих  $X_n = x_1, \dots, x_n$ .

Выделим в группе  $\text{Aut} L_n$  всех автоморфизмов алгебры  $L_n$  подгруппу  $U_n$ , порождённую автоморфизмами вида:

$$\tau_i(y_i) : \begin{cases} x_i \rightarrow x_i + y_i, \\ x_j \rightarrow x_j, \quad j \neq i, \end{cases}$$

где  $y_i$  принадлежит подалгебре, порождённой  $x_{i+1}, \dots, x_n$ . Такая подгруппа называется группой унитреугольных автоморфизмов алгебры  $L_n$ .

Для краткости будем записывать произвольный автоморфизм  $\varphi$  свободной алгебры  $L_n$  с множеством свободных порождающих  $X_n$  как  $\varphi = (f_1, f_2, \dots, f_n)$ , где  $\varphi(x_i) = f_i$ ,  $i = 1, \dots, n$ .

Тогда произвольное отображение вида:

$$\varphi = (x_1 + f_1(x_2, \dots, x_n), \dots, x_i + f_i(x_{i+1}, \dots, x_n), \dots, x_n), \quad (1)$$

где для любого  $i$  многочлен  $f_i(x_{i+1}, \dots, x_n) \in L_n$  определяет автоморфизм из  $U_n$ , и группа  $U_n$  состоит из всех таких автоморфизмов.

Выделим в группе  $U_n$  следующие подгруппы  $Z_\alpha$  ( $1 \leq \alpha \leq \omega$ ), состоящие из автоморфизмов вида  $(x_1 + f_1(x_{n-1}, x_n), x_2, \dots, x_n)$ , причём в одночленах многочлена  $f_1(x_{n-1}, x_n)$  элемент  $x_{n-1}$  встречается не более чем  $\alpha - 1$  раз.

Очевидно, что  $Z_\alpha \subseteq Z_{\alpha+1}$ .

**Теорема 1.** Подгруппа  $Z_1$  является центром группы  $U_n$ .

*Доказательство.* Возьмём произвольные автоморфизмы  $\varphi \in Z_1$  и  $\psi \in U_n$ . Согласно описанию группы  $Z_1$ , автоморфизм  $\varphi$  имеет вид

$$\varphi = (x_1 + g_1(x_n), x_2, \dots, x_n),$$

где  $g_1(x_n) = \lambda_1 x_n + \lambda_2 x_n^2 + \dots + \lambda_k x_n^k + \dots$

Пусть  $\psi$  имеет вид (1). Вычислим  $\varphi \circ \psi$  и  $\psi \circ \varphi$ . Имеем

$$\varphi\psi = (x_1 + f_1(x_2, \dots, x_n) + g_1(x_n), x_2 + f_2(x_3, \dots, x_n), \dots, x_n).$$

Далее

$$\psi\varphi = (x_1 + g_1(x_n) + f_1(x_2, \dots, x_n), x_2 + f_2(x_3, \dots, x_n), \dots, x_n).$$

Убеждаемся, что эти композиции равны. Это по определению означает, что  $\varphi \in Z(U_n)$ .

Теперь предположим, что  $\varphi = (x_1 + f_1, x_2, \dots, x_{i-1}, x_i + f_i, x_{i+1}, \dots, x_n) \in Z(U_n)$ . Возьмём автоморфизм  $\psi = (x_1 + x_i, x_2, \dots, x_n)$ . Тогда

$$\varphi\psi = (x_1 + x_i + f_1, x_2, \dots, x_{i-1}, x_i + f_i, x_{i+1}, \dots, x_n),$$

$$\psi\varphi = (x_1 + f_1 + x_i + f_i, x_2, \dots, x_{i-1}, x_i + f_i, x_{i+1}, \dots, x_n).$$

Таким образом, поскольку  $\varphi\psi = \psi\varphi$ , следовательно,  $f_i = 0$ .

Пусть теперь  $\varphi = (x_1 + f_1(x_i, x_n), x_2, \dots, x_n) \in Z_1$ . Возьмём автоморфизм  $\psi = (x_1, \dots, x_i + x_n, \dots, x_n)$ . Тогда

$$\varphi\psi = (x_1 + f_1(x_i + x_n, x_n), x_2, \dots, x_i + x_n, \dots, x_n),$$

$$\psi\varphi = (x_1 + f_1(x_i, x_n), x_2, \dots, x_i + x_n, \dots, x_n).$$

Таким образом,  $f_1(x_i + x_n, x_n) = f_1(x_i, x_n)$  или  $f_1(x_i + x_n, x_n) - f_1(x_i, x_n) = 0$ . Но в общем случае эта разность содержит несократимые одночлены. Следовательно,  $f_1(x_i, x_n) = f_1(x_n)$ . ■

**Теорема 2.** Подгруппы  $Z_\alpha$  ( $1 \leq \alpha \leq \omega$ ) составляют центральный ряд группы  $U_n$ .

*Доказательство.* Напомним, что в центральном ряде  $Z_k(U_n)/Z_{k-1}(U_n) = Z(U_n/Z_{k-1})$ .

Допустим, что для некоторого  $1 \leq \alpha < \omega$  множества  $Z_1 \subseteq \dots \subseteq Z_{\alpha-1}$  являются частью центрального ряда. Пусть  $\varphi = (x_1 + g_1(x_{n-1}, x_n), x_2, \dots, x_n) \in Z_\alpha$ , а  $\psi$  — произвольный унитарный автоморфизм вида (1).

Составляя их композиции, видим, что

$$\varphi\psi = (x_1 + f_1 + g_1(x_{n-1} + f_n, x_n), x_2 + f_2, \dots, x_n),$$

$$\psi\varphi = (x_1 + g_1(x_{n-1}, x_n) + f_1, x_2 + f_2, \dots, x_n).$$

Многочлен  $f_n = f_n(x_n)$  состоит из одночленов вида  $x_n^p$ . Значит, в многочлене  $g_1(x_{n-1} + f_n, x_n)$  по сравнению с  $g_1(x_{n-1}, x_n)$  появляются одночлены, в которых переменная  $x_{n-1}$  встречается меньшее число раз. Из вида  $\varphi$  следует, что одночлены в  $g_1(x_{n-1} + f_n, x_n)$  либо совпадают с одночленами в  $g_1(x_{n-1}, x_n)$ , либо содержат переменную  $x_{n-1}$  не более чем  $\alpha - 2$  раз. Следовательно, в  $U_n/Z_{\alpha-1}$  композиции  $\varphi\psi$  и  $\psi\varphi$  совпадают.

Теперь докажем, что других автоморфизмов в  $Z_\alpha$  нет. Проведём рассуждения, аналогичные доказательству предыдущей теоремы.

С помощью автоморфизма  $\psi = (x_1 + x_i, x_2, \dots, x_n)$  можно доказать, что в автоморфизме  $\varphi$  многочлены  $f_i = 0$  для любого  $i \geq 2$ .

С помощью автоморфизма  $\psi = (x_1, \dots, x_i + x_n, \dots, x_n)$  можно доказать, что в автоморфизме  $\varphi$  многочлен  $f_1$  не содержит переменных  $x_i$  для  $i \leq n - 2$ .

А рассуждения, приведённые в данном доказательстве выше, показывают, что при наличии в  $f_1$  несократимого одночлена с  $\alpha$  или более включениями переменной  $x_{n-1}$  многочлен  $g_1(x_{n-1} + f_n, x_n)$  по сравнению с  $g_1(x_{n-1}, x_n)$  в общем случае будет содержать одночлены, не зануляющиеся в  $U_n/Z_{\alpha-1}$ .

Таким образом, множество  $Z_\alpha$  также входит в центральный ряд при условии, что туда входит  $Z_{\alpha-1} \subseteq Z_\alpha$ .

Так как для  $Z_1$  утверждение уже доказано, по индукции получаем то, что и требовалось доказать. ■

## ЛИТЕРАТУРА

1. Кабанов А.Н. Гиперцентральная структура группы унитарных автоморфизмов свободной метабелевой алгебры Ли // Сиб. мат. ж. 2009. Т. 50, № 2. С. 329–333.
2. Кабанов А.Н. Центр группы унитарных автоморфизмов свободной алгебры Ли // Математические структуры и моделирование. 2014. № 3 (31). С. 57–61.
3. Кабанов А.Н. Верхний центральный ряд группы унитарных автоморфизмов свободной алгебры Ли // Математическое и компьютерное моделирование: материалы III Международной научной конференции (Омск, 12 ноября 2015 г.). 2015. С. 100–102.
4. Loday J.-L. Une version non commutative des algebres de Lie: Les algebres de Leibniz // Enseign. Math. 1993. V. 39. N. 3–4. P. 269–293.

**CENTER SERIES OF THE GROUP OF UNITRIANGULAR AUTOMORPHISMS  
OF A FREE LEIBNIZ ALGEBRA****A.N. Kabanov**

Ph.D. (Phys.-Math.), e-mail: m01kab@mail.ru

Dostoevsky Omsk State University

**Abstract.** The center series of the group of unitriangular automorphisms of a free Leibniz algebra over an arbitrary field is described.

**Keywords:** Leibniz algebra, unitriangular automorphism, hypercenter.

*Дата поступления в редакцию: 30.08.2017*

## РЁБЕРНАЯ СВЯЗНОСТЬ ВЕРШИН ПОЛИЭДРОВ

**Т.К. Рустюмов**

математик, пенсионер, e-mail: rtk777@mail.ru

**С.Т. Рустюмова**

математик, директор Центра, e-mail: svrustjumova@mail.ru

Центр обучения математике

**Аннотация.** Авторы в начале статьи на конкретном примере в 3-хмерном пространстве демонстрируют необычное и до сих пор неизвестное свойство векторов. Для векторных пространств любой конечной размерности вводится понятие «конический базис», на основе которого доказываются новые свойства векторных пространств. Понятие конического базиса и некоторые другие новые идеи, позволяют доказать рёберную связность вершин полиэдров.

**Ключевые слова:** векторное пространство, вектор, конус, полиэдральный конус, конечнопорождённый конус, политоп, полиэдр, неравенство.

### Введение

Построим конкретный пример множества из четырёх векторов  $\{V_4\} = \{x, y, z, (-t)\}$  в 3-хмерном пространстве (определение самих векторов см. под рис. 1), которые не имеют крайних гиперплоскостей (опор):

Обозначим в  $E^3$  через  $G_i(r, t)$  гиперплоскость, проходящую через два линейно независимых вектора  $r$  и  $t$  (и начало координат), с номером  $i$ .

В нашем случае мы имеем всего шесть возможных «крайних» гиперплоскостей  $G_1(x, y)$ ,  $G_2(x, z)$ ,  $G_3(y, z)$ ,  $G_4(x, (-t))$ ,  $G_5(y, (-t))$ ,  $G_6(z, (-t))$  для  $\{V_4\} = \{x, y, z, (-t)\}$ .

Покажем аналитически, что в нашем примере «крайние» гиперплоскости  $G_i(r, t)$  строго разделяют оставшиеся два вектора  $\{V_4\} \setminus \{r, t\}$ .

В самом деле, для конуса  $C = \text{cone}(x, y, z)$  и его «крайних» гиперплоскостей  $G_1(x, y)$ ,  $G_2(y, z)$ ,  $G_3(x, z)$  пусть сначала выбрано  $G_1(x, y)$ . Тогда нормалью этой «крайней» гиперплоскости будет оставшийся вектор  $z = (0, 0, 1)$ . Следовательно, имеем скалярные произведения  $(z, z) = 1 > 0$ , а  $(z, (-t)) = -1 < 0$ , т. е. оставшиеся вектора  $z$  и  $(-t)$  строго разделяются нашей «крайней» гиперплоскостью  $G_1(x, y)$ .

Аналогично показывается, что «крайние» гиперплоскости  $G_2(y, z)$  и  $G_3(x, z)$  строго разделяют оставшиеся два вектора, не входящие в определение соответствующей «крайней» гиперплоскости.

Для «крайних» гиперплоскостей  $G_4(x, (-t))$ ,  $G_5(y, (-t))$ ,  $G_6(z, (-t))$  пусть сначала выбрана «крайняя» гиперплоскость  $G_4(x, (-t))$ , обозначим её нормаль

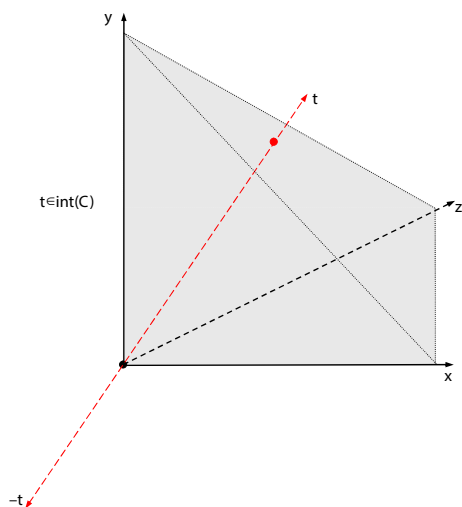


Рис. 1. Пример множества векторов  $\{V_4\} = \{x, y, z, (-t)\}$ , не имеющего «крайних» гиперплоскостей в 3-х мерном пространстве.

$x = (1, 0, 0)$ ,  $y = (0, 1, 0)$ ,  $z = (0, 0, 1)$ ,  
 $(-t) = (-1, -1, -1)$ ,  $C = \text{cone}(x, y, z)$

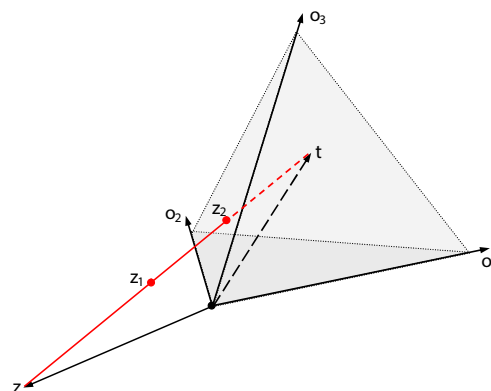


Рис. 2. Пояснение доказательства леммы для 3-х мерного пространства

через  $n_x = (a, b, c)$ , где хотя бы одна координата не нулевая. Тогда из скалярного произведения  $(n_x, x) = 0$ , следует  $(a) = 0$ , отсюда  $(b + c) = 0$  или  $(c) = (-b)$ . Итак,  $n_x = (0, b, -b)$  и мы можем принять  $n_x = (0, 1, -1)$ . Тогда скалярное произведение  $(n_x, y) = 1 > 0$ , а скалярное произведение  $(n_x, z) = -1 < 0$ , т.е. «крайняя» гиперплоскость  $G_4(x, (-t))$  строго разделяет оставшиеся два вектора  $y$  и  $z$ .

Аналогично показывается, что и «крайние» гиперплоскости  $G_5(y, (-t))$  и  $G_6(z, (-t))$  строго разделяют оставшиеся два вектора, не входящие в определение соответствующей «крайней» гиперплоскости.

То есть в этом случае, для примера, в  $E^3$  — **вообще нет «крайних» гиперплоскостей (опор) и невозможно построение полиэдрального конуса для набора векторов  $\{V_4\}$ .**

**Примечание** (визуальное моделирование). Рис. 1 легко смоделировать на углу прямоугольного стола. Надо взять два карандаша или две ручки. Один карандаш надо поставить вертикально на выбранном углу стола — этот карандаш и ребра угла стола будут определять координаты  $(x, y, z)$ . Вторым карандашом надо направить из выбранного угла стола в сторону вектора  $(-1, -1, -1)$  — это и будут четыре вектора с рис. 1. Теперь, как бы вы не проводили «крайнюю» гиперплоскость (в виде тонкой тетради или листа картона) через два любых линейно независимых вектора (и начало координат), эта гиперплоскость будет строго разделять два оставшихся вектора.

Будем придерживаться следующих, почти традиционных, обозначений и определений.

Через  $E^d$  — обозначим  $d$ -мерное векторное пространство, т. е. пространство  $d$ -плексов  $x = (x_1, x_2, \dots, x_d)$ , состоящее из действительных чисел  $x_i$ , где  $i = 1, \dots, d$ .

Будем называть  $\mathbf{d}$ -плекс **точкой** или **вектором** пространства  $E^d$ .

Начало координат, точку  $(0, 0, \dots, 0)$  будем обозначать через  $\mathbf{0}$ .

**Выпуклой комбинацией** векторов  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$  называется любой вектор  $\mathbf{v} = \sum_1^m \lambda_i \mathbf{v}_i$ , где  $\lambda_i \geq 0$ , и  $i \in 1, 2, \dots, m$  с дополнительным условием  $\sum_1^m \lambda_i = 1$ . Выпуклая комбинация векторов для кратности обозначается  $\text{conv}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$  и называется **выпуклой оболочкой** векторов  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ .

**Конической комбинацией** векторов  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$  называется любой вектор  $\mathbf{v} = \sum_1^m \lambda_i \mathbf{v}_i$ , где  $\lambda_i \geq 0$ , для  $i \in 1, 2, \dots, m$ . Коническая комбинация векторов  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$  для кратности обозначается  $\text{cone}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$  и называется **выпуклым конусом**, а вектора  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$  называются **образующими** конуса.

**Остовом конуса** называется минимальный набор его образующих, т.е. только те вектора  $\mathbf{v}_i$ , которые нельзя выразить через другие образующие в виде конической комбинации.

Конус называется **заострённым**, если он не содержит подпространств (кроме точки  $\mathbf{0}$ ).

Считаем  $\mathbf{d}$ -конусом в  $E^d$ , конус с  $\mathbf{d}$  линейно независимыми образующими  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$  и с вершиной в точке  $\mathbf{0}$ . Для нас  $\mathbf{d}$ -конус удобен тем, что он всегда полиэдральный ( $\mathbf{d}$ -конус является заострённым) и его гиперплоскости легко определяются через нормали к любым  $(\mathbf{d}-1)$  линейно независимым векторам его остова.

Для  $\mathbf{d}$ -конуса  $\mathbf{C}$  в  $E^d$ , имеющего остов из  $\mathbf{d}$  линейно независимых векторов  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d$ , обозначим через  $\mathbf{C}^- = \text{cone}(-\mathbf{v}_1, -\mathbf{v}_2, \dots, -\mathbf{v}_d)$ .

Конус называется **конечнопорождённым** векторами  $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$ , если он является конической комбинацией всех или некоторых векторов  $(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m)$  (это выпуклый конус). Конечнопорождённый конус в некоторых учебниках называют многогранным конусом, что не совсем корректно в случае конического базиса, который не имеет ни отделяющих, ни крайних гиперплоскостей (см. далее теорему 1 и следствие 2 из неё).

Конус называется **полиэдральным**, если он является пересечением конечного числа линейных замкнутых полупространств (т.е. не строгих неравенств), проходящих через  $\mathbf{0}$ , и содержащих не нулевые векторы.

Изолированную точку  $\mathbf{0}$  (начало координат) мы не будем считать ни полиэдральным ни конечнопорождённым конусом (см. пункт **Пояснение** в доказательстве теоремы 2).

Если  $\mathbf{G}$  гиперплоскость в  $E^d$  с нормалью  $\mathbf{n}$ , т.е.  $\mathbf{G} = \{\mathbf{x} \setminus (\mathbf{x}\mathbf{n} = 0)\}$ , то будем обозначать полупространство  $\mathbf{G}^+ = \{\mathbf{x} \setminus (\mathbf{x}\mathbf{n} \geq 0)\}$  и  $\mathbf{G}^- = \{\mathbf{x} \setminus (\mathbf{x}\mathbf{n} \leq 0)\}$ . Как обычно, в фигурных скобках указываем некоторые множества.

Размерность векторов  $\{\mathbf{V}\}$  обозначается  $\dim(\{\mathbf{V}\})$  и определяется как размерность минимального подпространства, содержащего вектора  $\{\mathbf{V}\}$ .

Символом  $||\{\mathbf{V}\}||$  обозначается количество векторов  $\{\mathbf{V}\}$ .

Аббревиатура СЛН является сокращением слов Система Линейных Неравенств.

Далее изложим некоторые новые свойства векторов в многомерных векторных пространствах конечной размерности, которые нам необходимы для доказательства рёберной связности вершин политопов и полиэдров.

## 1. Конические базисы и их основные свойства

**Определение 1.** Коническим базисом в  $E^d$  называется множество векторов  $\{V\} = \{v_1, v_2, \dots, v_m\}$ , такое, что любой вектор из  $E^d$  выражается через вектора  $\{V\}$  некой конической комбинацией (т.е. с неотрицательными коэффициентами).

Это определение означает, что  $\text{cone}(\{V\}) = E^d$  и  $\dim(\{V\}) = d$ .

**Лемма 1.** (О существовании конических базисов). В  $E^d$  существуют конические базисы.

*Доказательство.* Пусть в  $E^d$  заданы  $d$  линейно независимых векторов  $\{v_1, v_2, \dots, v_d\}$  (с началом в точке  $0$ ), которые образуют  $d$ -мерный полиэдральный конус  $C$  с образующими-остовом  $\{v_1, v_2, \dots, v_d\}$  и ровно  $d$  гипергранями. Пусть вектор  $t$  принадлежит  $\text{int}(C)$ .

Тогда для любого вектора  $z$ , не принадлежащего конусу  $C$  и  $z \neq \gamma t$ , где  $\gamma < 0$ , вектор  $z$  и  $t$  можно соединить отрезком  $[z, t]$  (ведь  $E^d$  тоже выпуклое множество), который пересечёт некоторую гипергрань конуса  $C$  в точке  $z_1$ , принадлежащей отрезку  $[z, t]$ .

Если  $z_1$  ещё не принадлежит выпуклому конусу  $C$ , то на отрезке  $[z, t]$  можно найти точку  $z_2$ , образованную пересечением некой другой гиперграни конуса  $C$ , и которая находится ближе к точке  $t$ , чем точка  $z_1$  (расстояния на отрезке определяются очень просто). Поскольку различных гиперграней у конуса  $C$  ровно  $d$ , то мы, в конце концов, найдём точку  $z_i$ , где  $i < d$ , которая принадлежит гиперграни конуса  $C$  и

$$z_i = \sum_1^d \beta_i v_i,$$

где  $\beta_i \geq 0$ . Тогда верно представление

$$\alpha z + (1 - \alpha)t = z_i = \sum_1^d \beta_i v_i,$$

где  $\beta_i \geq 0$  и  $1 > \alpha > 0$ , т. е.  $\alpha \neq 0$ , т. к.  $t \in \text{int}(C)$ . Это означает, что вектор  $z$  можно представить в виде

$$\alpha z = \sum_1^d \beta_i v_i + (1 - \alpha)(-t),$$

где  $1 > \alpha > 0$ , а это означает, что множество векторов  $\{v_1, v_2, \dots, v_d, (-t)\}$  по вышеприведённому определению является коническим базисом в  $E^d$ . ■



Свойства классических базисов, определяющих размерность векторного пространства, существенным образом отличаются от свойств конических базисов. Ясно, что если к исходному коническому базису  $\{\mathbf{V}\}$  добавить произвольное множество векторов, то получим новое множество — новый конический базис  $\{\mathbf{V}^+\}$  с большим количеством векторов, чем у первоначального  $\{\mathbf{V}\}$ .

Поэтому конические базисы, в отличие от линейно независимых векторов, могут содержать любое количество векторов, но их число должно быть строго больше, чем размерность исходного пространства.

**Такое свойство конических базисов вызывает затруднения при работе с ними, но некоторые трудности преодолимы и дают возможность получить важные результаты в теории линейных неравенств, которые без рассмотрения конических базисов не могли быть получены.**

Теперь докажем основное свойство конических базисов.

**Теорема 1.** (Об основном свойстве конических базисов). *Множество векторов  $\{\mathbf{V}\}$  ранга  $\mathbf{d}$  в  $\mathbf{E}^{\mathbf{d}}$  является коническим базисом тогда и только тогда, когда любая гиперплоскость, проходящая через начало координат, не является отделяющей гиперплоскостью для всех векторов из  $\{\mathbf{V}\}$ .*

*Доказательство. Необходимость:* Рассмотрим в  $\mathbf{E}^{\mathbf{d}}$  любую гиперплоскость  $\mathbf{G}$  (с замкнутыми полупространствами  $\mathbf{G}^+$  и  $\mathbf{G}^-$  и с нормалью  $\mathbf{n}_g$ ), проходящую через начало координат. Такая гиперплоскость  $\mathbf{G}$  не может быть отделяющей любого из своих двух полупространств  $\mathbf{G}^+$  или  $\mathbf{G}^-$  для всех векторов  $\{\mathbf{V}\}$ .

В самом деле, если какое-то полупространство ( $\mathbf{G}^+$  или  $\mathbf{G}^-$ ) является отделяющей гиперплоскостью для векторов  $\{\mathbf{V}\}$ , то в другом полупространстве гиперплоскости  $\mathbf{G}$ , были бы не нулевые вектора из  $\mathbf{E}^{\mathbf{d}}$  (например, одна из нормалей  $\mathbf{n}_g$  или  $(-\mathbf{n}_g)$ ), которые нельзя выразить через вектора  $\{\mathbf{V}\}$  с неотрицательными коэффициентами (т.е. в виде конической комбинации), что противоречит тому, что  $\{\mathbf{V}\}$  является коническим базисом.

Отсюда следует, что для любой гиперплоскости  $\mathbf{G}$ , в полупространствах ( $\mathbf{G}^+$  и  $\mathbf{G}^-$ ) должны быть вектора из  $\{\mathbf{V}\}$ , которые строго отделяются гиперплоскостью  $\mathbf{G}$ .

*Достаточность:* Из векторов  $\{\mathbf{V}\}$  выделим некоторый  $\mathbf{d}$ -конус  $\mathbf{C} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d)$ , что несложно сделать, выделив из матрицы векторов линейно независимые вектора, ведь  $\dim(\{\mathbf{V}\}) = \mathbf{d}$ .

Обозначим  $\mathbf{T} = \{\mathbf{V}\} \setminus \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$ , а  $\mathbf{R} = \text{conv}(\{\mathbf{T}\})$  — выпуклая оболочка векторов  $\mathbf{T}$ .

Рассмотрим  $\mathbf{S} = (\mathbf{C}^-) \cap \mathbf{R}$ . Если  $\mathbf{S}$  не содержит внутренних вектора из  $\mathbf{R}$  и  $\mathbf{C}^-$ , то выпуклые множества  $\mathbf{R}$  и  $\mathbf{C}^-$  можно отделить некой гиперплоскостью  $\mathbf{G}$  (здесь мы опираемся на теоретико-множественную теорему об отделимости выпуклых множеств, см., например, [5]). При этом можно считать, что гиперплоскость  $\mathbf{G}$  проходит через точку  $\mathbf{0}$ .

В самом деле, если  $\mathbf{G}$  проходит через некоторую грань конуса  $\mathbf{C}^-$  (но не содержит внутренних точек  $\mathbf{C}^-$ ), то гиперплоскость  $\mathbf{G}$  тоже проходит через  $\mathbf{0}$ , как и все грани  $\mathbf{C}^-$ .

Если же гиперплоскость  $\mathbf{G}$  строго отделяет конус  $\mathbf{C}^-$ , пересекая конус  $\mathbf{C}^+$ , то гипергрань  $\mathbf{G}$  можно параллельно сдвинуть в точку  $\mathbf{0}$ , при этом сдвинутая  $\mathbf{G}_0$  будет отделять  $\mathbf{C}^-$  и  $\mathbf{R}$ .

Действительно, если бы гиперплоскость  $\mathbf{G}_0$  содержала внутренние точки конуса  $\mathbf{C}^-$ , то  $\mathbf{G}_0$  обратно сдвинутая параллельно до гиперплоскости  $\mathbf{G}$ , продолжала бы пересекать конус  $\mathbf{C}^-$  по внутренним точкам, т. к. конус  $\mathbf{C}^-$  не ограничен.

Итак, мы можем считать отделяющую гиперплоскость  $\mathbf{G}$ , проходящей через начало координат. Хотя гиперплоскость  $\mathbf{G}$  может содержать какие-то граничные точки  $\mathbf{C}^-$  и  $\mathbf{R}$ , но по нашему предположению гиперплоскость  $\mathbf{G}$  не содержит внутренние точки  $\mathbf{R}$  и  $\mathbf{C}^-$ .

Допустим теперь,  $\mathbf{R}$  лежит в полупространстве  $\mathbf{C}^+$ , тогда  $\mathbf{C}^-$  лежит в полупространстве  $\mathbf{G}^-$ . А это означает, что  $\mathbf{d}$ -конус  $\mathbf{C}$  будет лежать в полупространстве  $\mathbf{G}^+$ . Отсюда следует, что вектора  $\mathbf{d}$ -конуса  $\mathbf{C} = \text{cone}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d)$  и выпуклой оболочки  $\mathbf{T} = \{\mathbf{V}\} \setminus \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$  лежат в одном полупространстве  $\mathbf{G}^+$ . То есть все вектора  $\{\mathbf{V}\}$  отделяются гиперплоскостью  $\mathbf{G}^+$ , а это противоречит условию теоремы.

Отсюда следует, что  $\mathbf{S} = (\mathbf{C}^-) \cap \mathbf{R}$  содержит внутренние точки как конуса  $\mathbf{C}^-$ , так и выпуклой оболочки  $\mathbf{R}$ . Пусть вектор  $\mathbf{w}$  является внутренней точкой  $\mathbf{C}^-$  и  $\mathbf{R}$ . Тогда вектор  $\mathbf{w}$  является выпуклой комбинацией векторов  $\mathbf{T} = \{\mathbf{V}\} \setminus \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d\}$  (частный случай конической комбинации) и одновременно  $\mathbf{w}$  принадлежит  $\mathbf{C}^-$ .

Это означает, что множество векторов  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_d, \mathbf{w}\}$ , где вектор  $\mathbf{w}$  является выпуклой комбинацией векторов  $\{\mathbf{V}\}$ , будет коническим базисом по лемме 1 (о существовании конических базисов). ■

**Следствие 1.** Если для множества векторов  $\{\mathbf{V}\}$  ранга  $\mathbf{d}$  можно найти хотя бы одну, проходящую через начало координат отделяющую гиперплоскость для векторов из  $\{\mathbf{V}\}$ , то это множество векторов  $\{\mathbf{V}\}$  не является коническим базисом.

**Следствие 2.** Если множество векторов  $\{\mathbf{V}\}$  ранга  $\mathbf{d}$  является коническим базисом, то любая гиперплоскость, проходящая через начало координат (в том числе и крайние гиперплоскости, проходящие через  $(\mathbf{d} - 1)$  линейно независимых вектора из  $\{\mathbf{V}\}$ ), не является отделяющей гиперплоскостью для векторов из  $\{\mathbf{V}\}$ . То есть в этом случае множество отделяющих (в том числе и крайних) гиперплоскостей для  $\{\mathbf{V}\}$  будет пустым множеством.

**Теорема 2.** (О возможности построения полиэдрального конуса векторов). Для множества векторов  $\{\mathbf{V}\}$  ранга  $\mathbf{d}$  в  $\mathbf{E}^d$  можно построить полиэдральный конус тогда и только тогда, когда множество векторов  $\{\mathbf{V}\}$  не является коническим базисом.

*Доказательство. Необходимость:* из теоремы-1 мы знаем, что для конических базисов нельзя построить отделяющую гиперплоскость, а значит и полиэдральный конус.

*Достаточность:* для множества векторов  $\{\mathbf{V}\}$ , не являющихся коническим базисом, возможно построение полиэдрального конуса.

Создадим СЛН

$$v_j x \leq 0, \quad j = 1, 2, \dots, m, \quad (1)$$

где нормальными неравенств являются вектора из  $\{\mathbf{V}\} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ .

Мы будем рассматривать СЛН (1) как пару двойственных конусов. Известно, что СЛН (1) всегда совместна, т. к. имеет решением хотя бы точку  $\mathbf{0}$ .

Но нас не устраивает случай, когда решением является только точка  $\mathbf{0}$ , мы не считаем одну точку  $\mathbf{0}$  полиэдральным конусом.

**Пояснение.** Здесь есть одна тонкость. Когда решением СЛН (1) является единственная точка  $\mathbf{0}$ , то вектора из  $\{\mathbf{V}\}$ , возможно, образуют конический базис (см. теорему 8.1 из [1]) и не могут иметь отделяющих гиперплоскостей (см. теорему 1 из данной статьи).

В особом случае, когда решения (1) образуют полиэдральный конус, содержащий не нулевые вектора, тогда для векторов  $\{\mathbf{V}\}$  можно построить полиэдральный конус (см. далее доказательство этой теоремы).

Мы покажем, что решением СЛН (1) является полиэдральный конус, содержащий не нулевые векторы, в случае, если  $\{\mathbf{V}\}$  не образуют конического базиса.

Поскольку вектора  $\{\mathbf{V}\}$  не образуют конического базиса, то по основной теореме 1 о конических базисах, существует отделяющая гиперплоскость  $\mathbf{G}$  с нормалью  $\mathbf{n}$ , проходящая через точку  $\mathbf{0}$  и отделяющая все вектора  $\{\mathbf{V}\}$ .

Так как вектора из  $\{\mathbf{V}\}$  лежат в другом полупространстве (в полупространстве  $\mathbf{G}^-$ ), чем нормаль  $\mathbf{n}$ , то нормаль этой отделяющей гиперплоскости  $\mathbf{n}$  будет решением СЛН (1).

Более того, эта нормаль  $\mathbf{n}$  с любым неотрицательным коэффициентом тоже является решением (1), и тогда система (1) имеет неограниченные решения — следовательно, существует выпуклый полиэдральный конус  $\mathbf{C}$  решений созданной СЛН (1), ведь конус  $\mathbf{C}$  является пересечением полупространств из СЛН (1). Размерность конуса  $\mathbf{C}$  может быть от 1 до  $\mathbf{d}$ .

Кроме того, этот выпуклый полиэдральный конус решений  $\mathbf{C}$  является заострённым. В самом деле, если  $\mathbf{C}_0$  является не нулевым подпространством  $\mathbf{C}$ , то множество векторов  $\{\mathbf{V}\}$  должно лежать в грани, ортогональной  $\mathbf{C}_0$ , имеющей размерность менее  $\mathbf{d}$ , т. е.  $\{\mathbf{V}\}$  должно иметь размерность менее  $\mathbf{d}$ , что противоречит условию теоремы.

Если существует выпуклый полиэдральный заострённый конус решений  $\mathbf{C}$ , построенной СЛН (1), то существует и его двойственной выпуклый конус  $\mathbf{C}^*$  — коническая оболочка нормалей СЛН (1), т. е. заданных векторов  $\{\mathbf{V}\}$ . Заострённый полиэдральный выпуклый конус  $\mathbf{C}$  имеет единственный остов, т. е. вектора-ребра полиэдрального конуса  $\mathbf{C}$ , коническими комбинациями которых образуется весь конус  $\mathbf{C}$ .

Каждое ребро остова конуса  $\mathbf{C}$  является нормалью крайней гиперплоскости выпуклого конуса  $\mathbf{C}^*$ , т. е. у конуса  $\mathbf{C}^*$  существуют крайние гиперплоскости, которые проходят через  $(\mathbf{d}-1)$  линейно независимых векторов из  $\{\mathbf{V}\}$ .

Отметим также, что даже если конус решений СЛН (1), т.е конус  $\mathbf{C}$  является всего лишь одним лучом  $\mathbf{r}$ , которое также будет и нормалью единственной гиперплоскости  $\mathbf{G}_r$  конуса  $\mathbf{C}^*$ , даже в этом случае гиперплоскость  $\mathbf{G}_r$ , соответствующая нормали  $\mathbf{r}$ , содержит  $(\mathbf{d}-1)$  линейно независимых векторов из  $\{\mathbf{V}\}$ , которые определяют гиперплоскость  $\mathbf{G}_r$ .

Только теперь можно просто перебирать наборы из  $(\mathbf{d}-1)$  линейно независимых векторов из  $\{\mathbf{V}\}$  и найти все крайние гиперплоскости  $\mathbf{C}^*$  (которые существуют, как это показано выше), но с одним уточнением. Выбранное множество из  $(\mathbf{d}-1)$  линейно независимых векторов, которое определяет крайнюю гиперплоскость  $\mathbf{G}_f$ , может содержать в гиперплоскости  $\mathbf{G}_f$  более чем  $(\mathbf{d}-1)$  вектор из  $\{\mathbf{V}\}$ , и эти вектора тоже могут участвовать в образовании крайней гиперплоскости  $\mathbf{G}_f$ .

Это означает, что может быть несколько различных наборов из  $(\mathbf{d}-1)$  линейно независимых векторов, которые будут определять одну и ту же крайнюю гиперплоскость  $\mathbf{G}_f$ .

Но нормаль  $\mathbf{n}_f$  как ребро остова конуса  $\mathbf{C}$  у этих крайних гиперплоскостей будет единственной, и именно нормаль  $\mathbf{n}_f$  позволит нам исключать не совпадающие множества из  $(\mathbf{d}-1)$  линейно независимых векторов, которые создают совпадающие крайние гиперплоскости с нормалью  $\mathbf{n}_f$ . ■

**Замечание.** Условие  $\dim(\{\mathbf{V}\}) = \mathbf{d}$  важно. Если  $\dim(\{\mathbf{V}\}) = \mathbf{k} < \mathbf{d}$  и  $\|\{\mathbf{V}\}\| > \mathbf{k}$ , то вектора  $\{\mathbf{V}\}$  в подпространстве  $\mathbf{E}^{\mathbf{k}}$  могут образовывать конический базис и не иметь отделяющих гиперплоскостей.

Теперь возникает естественный вопрос, как определить, что данное множество векторов  $\{\mathbf{V}\}$  является коническим базисом или нет (теорема 1 для этого не пригодна)?

**Теорема 3.** (О распознавании конического базиса). Если для множества векторов  $\{\mathbf{V}\}$  ранга  $\mathbf{d}$  в  $\mathbf{E}^{\mathbf{d}}$  любой набор из  $(\mathbf{d}-1)$  линейно независимых векторов из  $\{\mathbf{V}\}$  определяет гиперплоскость, которая не является отделяющей для векторов из  $\{\mathbf{V}\}$ , то множество векторов  $\{\mathbf{V}\}$  не является коническим базисом.

*Доказательство.* Будем доказывать от противного. Пусть множество векторов  $\{\mathbf{V}\}$  не является коническим базисом. Тогда по основной теореме 1 о конических базисах существует гиперплоскость, отделяющая вектора  $\{\mathbf{V}\}$ .

Но тогда по теореме 2 о возможности построения полиэдрального конуса векторов существует и крайняя гиперплоскость, проходящая через  $(\mathbf{d}-1)$  линейно независимых вектора для множества векторов  $\{\mathbf{V}\}$ . Это противоречие доказывает теорему. ■

**Замечание.** Конечно, теорему 3 можно было сформулировать в стиле «... тогда и только тогда, когда ...». но для нас было важно ответить на поставленный вопрос и при этом усилить теорему 1 так, чтобы можно было конечным перебором определить, является ли заданное множество векторов коническим базисом или нет.

## 2. Ребёрная связность вершин политопа и полиэдра

**Теорема 4.** совместная СЛН ранга  $d$  создаёт в  $E^d$  политоп (многогранник или ограниченный полиэдр) тогда и только тогда, когда нормали этой СЛН образуют конический базис.

*Доказательство.* Эта теорема доказана в [1], теорема 8.1, мы её переформулировали в терминах конических базисов для удобства использования в этой статье. ■

Для дальнейшего изложения нам необходимы некоторые новые обозначения и определения: Рассмотрим СЛН в  $E^d$ , где  $x^T = (x_1, x_2, \dots, x_d)$

$$n_j x \leq h_j \quad j = 1, 2, \dots, m, \quad (2)$$

где  $n_j$  – единичная нормаль гиперплоскости, определяемой неравенством с номером  $j$ , а  $h_j$  – опорное число неравенства  $j$ . Если СЛН (2) совместна, то совокупность решений (2) обозначим через  $P$ , для определённости считаем его  $d$ -мерным, это означает, что неравенства (гиперграни) каждой вершины  $v$  полиэдра  $P$ , проходящие через эту вершину  $v$ , образуют  $d$ -мерный заострённый конус  $C_v$ , т. к. конус содержит весь  $d$ -мерный  $P$  и имеет вершину  $v$ .

Множество  $P$  называется **политопом**, если он ограниченный, и **полиэдром**, если он неограниченный.

Политоп/полиэдр  $P$  имеет грани размерности от  $0$  (пересечение  $d$  линейно независимых неравенств) до  $(d-1)$ .

Грани  $P$  размерности  $0$  называются **вершинами**, а грани размерности  $1$  называются **рёбрами**.

Две вершины  $P$  называются **смежными**, если они соединены одним ребром.

Две вершины  $v$  и  $w$  называются **рёберно связными**, если существует последовательность смежных вершин, соединяющих вершины  $v$  и  $w$ .

Через  $N^d$  обозначим пространство для нормалей, которое абсолютно идентично  $E^d$ , но где все нормали собраны воедино и имеют начало в точке  $0$  (это облегчит понимание доказательства теорем о рёберной связности вершин полиэдров).

Через  $C_v^*$  обозначим конус, образованный нормальными неравенствами (2), проходящих через вершину  $v$ , и будем его называть **конусом нормалей вершины  $v$** , т. к. он находится в пространстве  $N^d$ . Конус  $C_v^*$  тоже будет заострённым и  $d$ -мерным, ведь он является двойственным к заострённому  $d$ -мерному конусу  $C_v$  в пространстве  $E^d$  в вершине  $v$ .

Через  $P^*$  обозначим совокупность всех нормалей  $d$ -мерного полиэдра  $P$  в  $N^d$ , определяемого неравенствами (2).

А для политопы  $P$ , как мы знаем из теоремы 4,  $P^*$  является коническим базисом в  $N^d$ .

Далее мы считаем, что СЛН (2) не имеет избыточных неравенств. Исключение избыточных неравенств можно осуществить для СЛН (2), например, по алгоритму Н.В. Черниковой из [4]. Кроме того, считаем, что политоп  $P$  имеет хотя бы две вершины (т. е. мы не рассматриваем однородные СЛН, которые

определяют конуса только с одной вершиной или когда решением СЛН является одна точка).

**Лемма 2.**  $P^*$  является выпуклым полиэдральным конусом для любого полиэдра.

*Доказательство.* Из теоремы 4 мы знаем, что только в случае политопа нормали неравенств образуют конический базис. Значит, в случае полиэдра, у нас нет конического базиса. Тогда по теореме 2 для нормалей полиэдра, т. е. для  $P^*$ , можно построить выпуклый полиэдральный конус. ■

**Теорема 5.** (О свойствах двойственных конусов различных вершин политопа и полиэдра). Пусть  $v$  и  $w$  две различные вершины политопа или полиэдра  $P$ . Тогда  $\text{int}(C_v^*) \cap \text{int}(C_w^*) = \emptyset$ . Для смежных вершин  $v$  и  $w$  будет  $\dim((C_v^*) \cap (C_w^*)) = (d - 1)$ .

*Доказательство.* Сместим двойственные конуса  $C_v^*$  и  $C_w^*$  из начала координат пространства  $N^d$  в соответствующие вершины  $v$  и  $w$  политопа или полиэдра  $P$  в основном пространстве  $E^d$ . Отрезок  $[v-w]$  целиком лежит в  $P$  (свойство выпуклости  $P$ ).

Поэтому гиперплоскость  $G_v$  с нормалью  $[w-v]$ , проходящая через вершину  $v$ , отделяет  $C_v^*$ , смещённую в точку  $v$ . Это обусловлено тем, что нормаль  $[w-v]$  лежит в двойственном конусе для  $C_v^*$  в основном пространстве  $E^d$ . Так же гиперплоскость  $G_w$  с нормалью  $[v-w]$ , проходящая через вершину  $w$ , отделяет  $C_w^*$ , смещённую в точку  $w$ . И в этом случае нормаль  $[v-w]$  лежит в двойственном конусе для  $C_w^*$  в основном пространстве  $E^d$ .

Поскольку гиперплоскости  $G_v$  и  $G_w$  имеют противоположно направленные нормали, то гиперплоскости  $G_v$  и  $G_w$ , смещённые в точку  $0$  в  $N^d$ , совпадут, и эта совмещённая в  $0$  гиперплоскость будет строго отделять открытые конуса  $\text{int}(C_v^*)$  и  $\text{int}(C_w^*)$  в пространстве  $N^d$ .

Если же вершины  $v$  и  $w$  являются смежными для политопа или полиэдра, т. е. соединены одним конечным ребром  $r$  политопа или полиэдра  $P$ , то пересечение  $C_v^*$  и  $C_w^*$  в пространстве  $N^d$  будут содержать ровно  $(d-1)$  линейно независимые нормали, определяющие это конечное ребро  $r$ . Ребро определяется ровно  $(d-1)$  линейно независимой нормалью, т. к. нет избыточных неравенств (т. е. в этом случае соответствие взаимно однозначное).

Эта теорема верна как для политопов, так и для полиэдров. ■

Из теоремы 4, леммы 2 и теоремы 5 следует важный вывод:  $P^*$  любого политопа или полиэдра в  $N^d$  имеет плотно упакованную (без пропусков и «щелей») «сотовую» структуру конусов нормалей вершин в пространстве  $N^d$ .

Внутренние части конусов нормалей  $\text{int}(C_v^*)$  и  $\text{int}(C_w^*)$  не смежных или смежных вершин  $P^*$  не пересекаются, и мы считаем их соответствующими вершинам  $v$  и  $w$  в основном пространстве  $E^d$ , и это соответствие взаимно однозначное: вершина  $v \Leftrightarrow \text{int}(C_v^*)$ ; вершина  $w \Leftrightarrow \text{int}(C_w^*)$ .

Но конуса нормалей вершин у смежных вершин  $P^*$  плотно соприкасаются друг с другом своими гипергранями, определяемыми  $(d-1)$  линейно независимыми нормальными конечного ребра, соединяющего эти смежные вершины политопа  $P$  (если нет избыточных неравенств, то это соответствие тоже взаимно однозначное).

Именно такое соответствие и такая сотовая структура  $P^*$  позволяет нам получить крайне важный и чисто геометрический результат о рёберной связности любых вершин политопа или полиэдра  $P$ .

**Теорема 6.** *Любые две различные вершины политопа являются рёберно связными.*

*Доказательство.* Пусть заданы две не смежные  $v$  и  $w$  вершины политопа  $P$ . Тогда по теореме 5 внутренние части конусов вершин  $C_v^*$  и  $C_w^*$  не пересекаются, хотя сами конуса этих двух вершин могут иметь строго менее  $d-1$  общих нормалей, т. к. вершины  $v$  и  $w$  не являются смежными.

Тогда строго внутри конусов вершин  $C_v^*$  и  $C_w^*$  можно найти не нулевые векторы  $r$  и  $t$  соответственно так, чтобы  $r \neq \lambda t$ , где  $\lambda < 0$  (т. е. отрезок  $[r, t]$  не должен проходить через начало координат, т. е. через точку  $0$ ).

В теореме 5 мы установили плотную сотовую структуру конического базиса  $P^*$  пространства  $N^d$ . Кроме того, установили соответствие между вершинами и рёбрами заданного полиэдра  $P$  с соответствующими частями  $P^*$ .

Это позволяет нам утверждать, что отрезок  $[r, t]$  будет пересекать конуса вершин без возвратов (т. е. нет циклов в терминах теории графов) и повторений конусов вершин, т. к. построенный отрезок  $[r, t]$  является отрезком прямой.

Ясно что, когда отрезок прямой  $[r, t]$  какими-то своими точками находится внутри некоторого конуса вершины  $s(P^*)$ , т. е. внутри  $C_s^*$ , то он «определяет» этими точками соответствующую вершину  $s(P)$  исходного политопа  $P$ . А когда отрезок  $[r, t]$  пересекает гипергрань  $G_s$  двойственной вершины  $C_s^*$ , то этот отрезок  $[r, t]$  определяет на исходном полиэдре  $P$  ребро (т. е. такое существует), исходящее из вершины  $s(P)$ .

Пусть отрезок  $[r, t]$  будет пересекать не гипергрань  $C_s^*$ , которая должна иметь размерность  $(d-1)$ , а некоторую грань  $K_s$  меньшей размерности у конуса  $C_s^*$  текущей вершины  $s(P^*)$ . Тогда можно брать любую гипергрань  $C_k^*$ , определяемую  $(d-1)$  линейно независимыми нормальными, которая содержит эту грань  $K_s$  (т. е. и в этом случае существует ребро  $P$ , по которому можно продолжить движение).

В самом деле, поскольку отрезок  $[r, t]$  пересекает грань  $K_s$ , то отрезок  $[r, t]$  будет пересекать и любую гипергрань  $C_k^*$ , содержащую грань  $K_s$ . А это означает, что любая гипергрань  $C_k^*$ , содержащая грань  $K_s$ , не мешает дальнейшему движению вдоль отрезка  $[r, t]$  к целевой вершине  $w$ , т. е. гипергрань  $C_k^*$  никоим образом не «экранирует» движение к вершине  $w$ .

Таким образом, построенный в коническом базисе  $P^*$  отрезок  $[r, t]$  генерирует на исходном политопе  $P$  некоторую простую цепь (в терминах теории графов) рёберно связанных вершин, соединяющих выбранные вершины  $v$  и  $w$ .

При этом простая цепь, сгенерированная на политопе  $\mathbf{P}$ , скорее всего, будет зигзагообразной. ■

Эта теорема важна для обоснования симплекс-метода, который был предложен Д.Б. Данцигом в 1949 г. для решения задач ЛП (линейного программирования). Однако предложенный Д.Б. Данцигом алгоритм, был чисто эвристическим (т. к. не было ещё доказательства рёберной связности вершин политопов и полиэдров).

Но в 1961 г. в теории графов появляется фундаментальная теорема Балинского (« $\mathbf{d}$ -многогранник является  $\mathbf{d}$ -связным»), хотя только для многогранников (т. е. только для политопов, т.к. для полиэдров теорема Балинского не применима).

Однако теорема Балинского в своём доказательстве опиралась на довольно сложную теорему Уитни из теории графов (доказательство теоремы Уитни можно посмотреть в [2]), которая в многочисленных простых учебниках по теории графов вообще не рассматривается.

Поэтому теорема Балинского обычно не использовалась при преподавании теории ЛП. При этом без каких-либо внятных обоснований в теории ЛП требовалась, чтобы все ресурсы задачи ЛП были ограниченными, т. к. тогда область допустимых решений будет многогранником (ограниченным полиэдром) и будет обеспечена рёберная связность вершин многогранника по теореме Балинского.

Теперь в преподавании теории ЛП можно приводить технически более простое и «наглядное» геометрическое доказательство рёберной связности вершин политопа (многогранника) для обоснования симплекс-метода ЛП без обращения к теории графов. В этом заключена ещё одна необходимость введения конических базисов.

**Теорема 7.** Любые две различные вершины полиэдра являются рёберно связными.

*Доказательство.* По лемме 2, в случае полиэдра, мы имеем в качестве  $\mathbf{P}^*$  выпуклый полиэдральный конус. Это означает, что отрезок  $[\mathbf{r}, \mathbf{t}]$ , выбранный строго внутри для конусов вершин  $\mathbf{v}$  и  $\mathbf{w}$ , будет лежать строго внутри выпуклого полиэдрального конуса  $\mathbf{P}^*$  и не может проходить через начало координат, т. е. через точку  $\mathbf{0}$ . Поэтому отрезок  $[\mathbf{r}, \mathbf{t}]$ , как и в теореме 6, будет генерировать на полиэдре  $\mathbf{P}$  некоторый путь, состоящий из вершин и ребер  $\mathbf{P}$  (возможно, зигзагообразный), который соединяет вершины  $\mathbf{v}$  и  $\mathbf{w}$ . ■

Эта важная теорема, которая опирается на лемму 2 и теорему 6, долго не поддавалась доказательству.

Теорема 7 тоже имеет большое значение в теории ЛП (например, для стандартных задач ЛП и некоторых двойственных задач ЛП, где нет ограничений на значения некоторых ресурсов).



**Следствие 3.** Любые две различные вершины как политопы, так и полиэдра являются рёберно связными.

Пояснение. Это следствие 3 является объединением результатов двух теорем: теоремы 6 и теоремы 7, которые несколько различны по доказательству, но сходны по смыслу.

## ЛИТЕРАТУРА

1. Ашманов С.А. Линейное программирование. М.: Наука. Гл. ред. физ.-мат. лит., 1981. 304 с.
2. Зыков А.А. Теория конечных графов. Т.1. СОАН Новосибирск. Наука, 1969. 543 с.
3. Схрейвер А. Теория линейного и целочисленного программирования. Т. 1. М. : Мир, 1991. 360 с.
4. Черников С.Н. Линейные неравенства. М. : Наука, 1968. 488 с.
5. Лейхтвейс К. Выпуклые множества. М.: Наука. Гл. ред. физ.-мат. лит., 1988. 335 с.

## EDGE CONNECTIVITY OF POLYHEDRA VERTICES

**T.K. Rustjumov**

Mathematician, Pensioner, e-mail: rtk777@mail.ru

**S.T. Rustjumova**

Mathematician, Director of Center, e-mail: svrustjumova@mail.ru

Center Mathematical Teaching

**Abstract.** At the beginning of the article the authors show unusual and still unknown property of vectors by a concrete example in the 3-dimensional space. The concept of "conical basis" is introduced upon which new properties of vector space are proved for vector spaces of any finite dimension. The concept of the conical base and some other new ideas, allow us to prove the connection rib vertex polyhedra.

**Keywords:** vector space, vector, cone, polyhedral cone, finitely generated cone, polytope, polyhedra, inequality.

*Дата поступления в редакцию: 24.08.2017*

## EXPERIMENTALLY OBSERVED DARK MATTER CONFINEMENT CLARIFIES A DISCREPANCY IN ESTIMATING THE UNIVERSE'S EXPANSION SPEED

**Olga Kosheleva**

Ph.D. (Phys.-Math.), Associate Professor, e-mail: [olgak@utep.edu](mailto:olgak@utep.edu)

**Vladik Kreinovich**

Ph.D. (Phys.-Math.), Professor, e-mail: [vladik@utep.edu](mailto:vladik@utep.edu)

University of Texas at El Paso, El Paso, Texas 79968, USA

**Abstract.** It is well known that our Universe is expanding. In principle, we can estimate the expansion speed either directly, by observing the current state of the Universe, or indirectly, by analyzing the cosmic background radiation. Surprisingly, these two estimates lead to somewhat different expansion speeds. This discrepancy is an important challenge for cosmologists. Another challenge comes from recent experiments that show that, contrary to the original idea that dark matter and regular (baryonic) matter practically do not interact, dark matter actually “shadows” the normal matter.

In this paper, we show that this “dark matter confinement” can explain the discrepancy between different estimates of the Universe’s expansion speed. It also explains the observed ratio of dark matter to regular matter.

**Keywords:** dark matter, Universe’s expansion speed.

### 1. Dark Matter: What Is It, and What Is Known About It

**Newton’s law of gravitation: brief reminder.** One of the main discoveries of Isaac Newton was that the motion of each celestial body  $A$  is described by the equation

$$m_A \cdot \vec{a}_A = \vec{F}_A = \sum_B \vec{F}_{AB} = \sum_B G \cdot \frac{m_A \cdot m_B}{r_{AB}^3} \cdot \vec{r}_{AB},$$

where  $m_A$  is the mass of the body  $A$ ,  $\vec{a}_A$  is its acceleration,  $G$  is the gravitational constant,  $\vec{r}_{AB}$  is a vector connecting the celestial bodies, and  $r_{AB}$  is the length of this vector, i.e., the distance between the bodies  $A$  and  $B$ .

By dividing both sides of this formula by  $m_A$ , we conclude that

$$\vec{a}_A = \sum_B G \cdot \frac{m_B}{r_{AB}^3} \cdot \vec{r}_{AB}.$$

By observing the motion of different celestial bodies, we can calculate their accelerations and thus, determine the corresponding masses.

In particular, by observing planets orbiting a star, we can determine the mass of this star; by observing satellites orbiting a planet, we can determine the mass of this planet, etc.

**What is dark matter.** In many galaxies, all the stars rotate around the galaxy's center (and, by the way, our Sun is no exception). In principle, by observing the acceleration of stars at different distance  $r$  from the galaxy's center, we can determine the overall mass of all the matter at distance  $\leq r$  from the galaxy's center.

On the other hand, we know what the galaxies are made of, so we can estimate the same mass by counting different stars — and adding up their masses, by counting the gases (which can be estimated by how much they emit or block the light from the stars), by estimating how many dead stars and planets there can be, etc.

It turns out that all possible matter constitutes only about 20% of the mass needed to explain the rotation. In other words, to explain the observed rotation, we need to conclude that, in addition to the usual mass, there is an additional “unusual” mass, mass that does not emit light or radio-waves and which is therefore not directly visible on the usual astronomical instruments such as telescopes and radio-telescopes. In a telescope, the area filled only with such an unusual matter is invisible (dark). Because of this fact, such unusual matter is called the *dark matter*.

*Comment.* To avoid possible confusion, we should mention that a similar deficiency appears if we compare the acceleration of galaxies on a cosmological scale with the overall mass of galaxies and other celestial bodies — the only difference is that on the cosmological scale, one needs to use Einstein's equations of General Relativity Theory instead of the approximate Newton's equations. The resulting missing mass is known as the *dark energy*.

In this paper, we only consider dark matter.

**What is the proportion of usual vs. dark matter.** According to current estimates, the usual matter forms  $\approx 5\%$  of the Universe's mass and dark matter forms  $\approx 27\%$  [4].

**The original idea: dark matter and usual matter practically do not interact.** Since the only observable effect of dark matter is its gravitational effect on the usual matter, researchers previously concluded that there is no other interaction between these two types of matter.

In other words, the only interaction between these two types of matter is the gravitational one — which, by the way, is drastically weaker than any other type of interaction; see, e.g., [1].

Because the only interaction between the two types of matter is very weak, it was assumed that, in general, these two types of matter practically do not affect each other. In particular, it was assumed that the spatial distributions of the usual matter and of the dark matter are very weakly correlated.

**An unexpected empirical fact: dark matter follows the usual matter.** Sur-

prisingly, a recent spatial analysis of galaxy's velocity and usual mass distributions showed that there is a strong spatial correlation between the usual matter and the dark matter [3].

While on the local level, the distributions of the dark matter may be practically independent from the distribution of the usual matter, on a larger scale, the distribution of the dark matter seems to follow the distribution of the usual matter — and vice versa.

In other words, while on a medium-size distance scale, there may be no interaction between the usual and the dark matter, something prevents them from separating beyond a certain distance.

## 2. This Experimentally Observed Dark Matter Confinement Can Clarify a Discrepancy Between Estimates of the Universe' Expansion Speed

**From the physical viewpoint, such a “confinement” is not unusual.** The above phenomenon is not unusual — a similar *quark confinement* is a known fact; see, e.g., [1]. Within short distances, quarks within a baryon practically do not interact with each other, but once they reach a certain distance, they become more and more attracted to each other — to the extent that they cannot separate beyond a certain distance.

**The discrepancy in the Universe's expansion speed.** While all observations confirm that the Universe expands, different measurements lead to different values of the expansion rate:

- supernova and quasar measurements estimate the expansion rate as 72–73 km/sec per megaparsec; see, e.g., [1], while
- measurements of the cosmic microwave background result in the smaller expansion rate of 67 km/sec per megaparsec [5].

From the physical viewpoint, cosmic background radiation measurements correspond to early stages of the Universe, while supernova and quasar measurements reflect the current state of the Universe. Thus, the above discrepancy means that at present, the Universe expands faster than in the past.

Until now, there was no good theoretical explanation for this discrepancy.

**Dark matter confinement can explain this discrepancy.** The observed phenomenon of dark matter confinement shows that at reasonably short distances, there is a strong attraction force between dark matter and regular (baryonic) matter.

At present, the confinement distance — which is commensurable with the galaxy sizes — is much smaller than the size of the Universe. Thus, these forces do not affect processes on the cosmological level of the Universe as a whole. On the other hand, on the early stages of the Universe, the distance was commensurable with the Universe's size. Thus, this additional attraction force prevented the Universe from expanding — and, as a consequence, the Universe expanded slower than at present.

This is exactly what we observe.

### 3. From Qualitative to Quantitative Explanation

**Need for quantitative explanation.** Qualitative explanation is a good start, but it is definitely desirable to come up with a quantitative explanation of observations. In this section, we provide such an explanation.

**Dark matter confinement explains the discrepancy between different estimates of the Universe's expansion speed.** In the dark matter confinement model, at the early stages of the Universe, there is a strong attraction between dark matter and regular matter, so strong that it prevents the dark matter from moving beyond a certain distance.

In effect, this means that at these stages, the dark and regular matter do not contribute to the expansion. Overall, both types of matter form  $27\% + 5\% \approx 32\%$  part of the Universe's mass. Since the gravitational interaction is proportional to the product of the interacting masses, the interaction between dark matter and normal matter forms  $0.32 \times 0.32 \approx 0.10$  part of the overall force. Thus, it is reasonable to expect that the expansion speed measured at the early stage of the Universe — e.g., by the cosmic background radiation — is approximately 10% smaller than the speed measured at present — e.g., by supernova and quasars.

A 10% decrease from 73 km/sec gives us 66 km/sec, which is in good accordance with the observed value of  $\approx 67$  km/sec per megaparsec.

### 4. Additional Result: Dark Matter Confinement Explains, in the First Approximation, the Observed Ratio of Dark Matter to Regular Matter

We know that dark matter “shadows” the regular matter. What is the configuration of dark matter in comparison with the usual matter?

- To each particle  $x$  of the normal matter, there can be a particle of dark matter nearby.
- We can also have several other particles of dark matter at some distance from the original particle  $x$ .

In nature, everything tries to move to a configuration with the smallest value of potential energy. Energy does not change if we swap different distance dark-matter particles. Thus, if we assume that there is one optimal configuration, then this optimal configuration should have the same symmetry as the energy expression. In other words, the configuration should not change if replace some of these distance dark-matter particles. This means, in particular, that the distance between every two different dark-matter particles must be the same.

In the 3-D space, we can have no more than 4 points  $A_1, A_2, \dots$ , with the property that  $d(A_i, A_j) = \text{const}$  for all  $i \neq j$ . These points form a regular tetrahedron.

In the resulting configuration, to every particle of the usual matter, there correspond *five* dark-matter particles:

- one close one and
- four distance ones.

This first-approximation ratio of dark matter to normal matter is in good accordance with the observed ratio of  $27\%/5\% \approx 5.4$ .

## Acknowledgments

This work was supported by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award “UTEP and Prudential Actuarial Science Academy and Pipeline Initiative” from Prudential Foundation.

## REFERENCES

1. Bonvin V. et al. H0LiCOW — V. New COSMOSGRAIL time delays of HE 0435–1223:  $H_0$  to 3.8 per cent precision from strong lensing in a flat  $\Lambda$ CDM model // Monthly Notices of the Royal Astronomical Society. 2017. Vol. 465. P. 4914–1930.
2. Feynman R., Leighton R., Sands M. The Feynman Lectures on Physics. Addison Wesley, Boston, Massachusetts, 2005.
3. McGaugh S.S., Lelli F., Schombert J.M. Radial acceleration relation in rotationally supported galaxies // Physical Review Letters. 2016. Vol. 117. P. 201101.
4. Planck Collaboration. Planck 2013 results. I. Overview of products and scientific results // Astronomy and Astrophysics. 2014. Vol. 571. P. A1.
5. Planck Collaboration. Planck intermediate results. XLVI. Reduction of large-scale systematic effects in HFI polarization maps and estimation of the reionization optical depth // Astronomy and Astrophysics. 2016. Vol. 596. P. A107.

## ЭКСПЕРИМЕНТАЛЬНО НАБЛЮДАЕМЫЙ КОНФАЙНМЕНТ ТЕМНОЙ МАТЕРИИ ПРОЯСНЯЕТ НЕСООТВЕТСТВИЕ В ОЦЕНКЕ СКОРОСТИ РАСШИРЕНИЯ ВСЕЛЕННОЙ

**О. Кошелева**

к.ф.-м.н., доцент, e-mail: [olgak@utep.edu](mailto:olgak@utep.edu)

**В. Крейнович**

к.ф.-м.н., профессор, e-mail: [vladik@utep.edu](mailto:vladik@utep.edu)

Техасский университет в Эль Пасо, США

**Аннотация.** Хорошо известно, что наша Вселенная расширяется. В принципе, мы можем оценить скорость расширения либо напрямую, наблюдая текущее состояние Вселенной, либо косвенно, анализируя космическое фоновое излучение.

Удивительно, но эти две оценки приводят к несколько разным скоростям расширения. Это несоответствие является важной задачей для космологов. Ещё одна проблема связана с недавними экспериментами, которые показывают, что, вопреки первоначальной идее, что тёмная материя и обычная (барионная) материя практически не взаимодействуют, тёмная материя фактически «затеняет» обычную материю. В этой статье мы показываем, что этот «конфайнмент тёмной материи» может объяснить несоответствие между различными оценками скорости расширения Вселенной. Это также объясняет наблюдаемое соотношение тёмной материи к обычной материи.

**Ключевые слова:** тёмная материя, скорость расширения Вселенной.

*Дата поступления в редакцию: 21.03.2017*

## DERIVATION OF GROSS-PITAEVSKII VERSION OF NONLINEAR SCHROEDINGER EQUATION FROM SCALE INVARIANCE

**Olga Kosheleva**

Ph.D. (Phys.-Math.), Associate Professor, e-mail: [olgak@utep.edu](mailto:olgak@utep.edu)

**Vladik Kreinovich**

Ph.D. (Phys.-Math.), Professor, e-mail: [vladik@utep.edu](mailto:vladik@utep.edu)

University of Texas at El Paso, El Paso, Texas 79968, USA

**Abstract.** It is known that in the usual 3-D space, the Schroedinger equation can be derived from scale-invariance. In view of the fact that, according to modern physics, the actual dimension of proper space may be different from 3, it is desirable to analyze what happens in other spatial dimensions  $D$ . It turns out that while for  $D \geq 3$  we still get only the Schroedinger's equation, for  $D = 2$ , we also get the Gross-Pitaevskii version of a nonlinear Schroedinger equation that describes a quantum system of identical bosons, and for  $D = 1$ , we also get a new nonlinear version of the Schroedinger equation.

**Keywords:** scale-invariance, nonlinear Schroedinger equation, Gross-Pitaevskii equation, system of identical bosons.

### 1. Formulation of the Problem

**Schroedinger's equation: a brief reminder.** In non-relativistic quantum mechanics, a state of a particle is described by a complex-valued wave function  $\psi(x, t)$ . The observational meaning of the wave function is that for each spatial location region  $\Omega$ , the probability to find the particle in this region is equal to  $\int_{\Omega} |\psi(x, t)|^2 dx$ ; see, e.g., [1].

The dynamics of the wave function is described, in the non-relativistic approximation, by the Schroedinger equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \cdot \nabla^2 \psi + V(x, t) \cdot \psi(x, t),$$

where:

- $i \stackrel{\text{def}}{=} \sqrt{-1}$ ,
- $\hbar$  is Planck's constant,
- $m$  is the particle's mass,
- $\nabla \stackrel{\text{def}}{=} \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_D} \right)$ , and
- $V(x, t)$  is the potential energy of the particle at location  $x$ .



This equation can be derived from the minimum action principle. Namely, the Schroedinger equation is equivalent to requiring that the field  $\psi(x, t)$  minimizes the action  $S \stackrel{\text{def}}{=} \int L(x, t) dx dt$ , where the function  $L$  – called the *Lagrange function* – has the form

$$L = i \cdot \hbar \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{\hbar^2}{2m} \cdot (\nabla \psi \cdot \nabla \psi^*) - V \cdot \psi \cdot \psi^*,$$

where:

- $\psi^*$  means complex conjugation, and
- for every two vectors  $\vec{a} = (a_1, \dots, a_D)$  and  $\vec{b} = (b_1, \dots, b_D)$ , the notation  $\vec{a} \cdot \vec{b}$  describes their dot (scalar) product  $\vec{a} \cdot \vec{b} = \sum_{i=1}^D a_i \cdot b_i$ .

**Schroedinger's equation can be derived from scale invariance.** In modern physics, the notions of symmetry play a fundamental role; see, e.g., [1, 6]. This makes perfect sense, since:

- the main purpose of science is to make predictions, and
- the only way we can make predictions about new situations is when we find some similarity (symmetry) between the new situations and situations that have been previously observed — and for which we know what happened.

One of the simplest symmetries comes from the fact that while physical equations deal with the numerical values of the physical quantities, these numerical values depend on the choice of the corresponding measuring units. If we use a new measuring unit which is  $\lambda$  times smaller than the previously used one, then all the numerical values of the corresponding quantity get multiplied by  $\lambda$ :  $x \rightarrow x' = \lambda \cdot x$ . For example, if we replace 1 m with 1 cm as the unit of length, then instead of 2 m, we get  $100 \cdot 2 = 200$  cm.

It is reasonable to require that the fundamental physical equations should not change if we simply re-scale the numerical values by changing the measuring units. It turns out that many fundamental physical equations — including Maxwell's equation for electrodynamics, Einstein's equation for General Relativity, and Schroedinger's equation of quantum mechanics — can be derived from this requirement of scale-invariance — plus a few other reasonable symmetries; see, e.g., [2, 3, 7, 8].

**What if we take into account that the dimension of proper space may be different from 3?** The above derivations deal with the usual 4-dimensional space-time, in which the proper space is 3-dimensional. However, according to modern physics, the actual dimension  $D$  of proper space may be different from 3; see, e.g., [4].

It is therefore desirable to analyze what happens if we look for scale-invariant equations and Lagrange functions in spatial dimensions  $D \neq 3$ .

**What we do in this paper.** In this paper, we show that for dimensions  $D \geq 3$ , we still get only the Schroedinger equation, but for  $D = 2$  and  $D = 1$ , we also get additional nonlinear versions of Schroedinger's equations:

- For  $D = 2$ , we also get the Gross-Pitayevskii equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \cdot \nabla^2 \psi + V(x, t) \cdot \psi(x, t) + \frac{c}{m} \cdot |\psi|^2 \cdot \psi,$$

where  $c$  is a constant. This equation describes a quantum system of identical bosons; see, e.g., [5, 9, 10]. This equation corresponds to the Lagrange function

$$L = i \cdot \hbar \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{\hbar^2}{2m} \cdot (\nabla \psi \cdot \nabla \psi^*) - V \cdot \psi \cdot \psi^* + \frac{f}{m} \cdot |\psi|^4.$$

- For  $D = 1$ , we also get the following new nonlinear version of the Schroedinger's equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \cdot \nabla^2 \psi + V(x, t) \cdot \psi(x, t) + \frac{c}{m} \cdot |\psi|^4 \cdot \psi.$$

This equation corresponds to the Lagrange function

$$L = i \cdot \hbar \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{\hbar^2}{2m} \cdot (\nabla \psi \cdot \nabla \psi^*) - V \cdot \psi \cdot \psi^* + \frac{f}{m} \cdot |\psi|^6.$$

## 2. Analysis of the Problem

**Lagrange function for non-relativistic quantum mechanics: a general description.** We want to obtain a Lagrange function describing the dynamics of a particle of mass  $m$ , described by a (complex-valued) wave function  $\psi(x, t)$ , in a field with a potential energy function  $V(x, t)$ . Since the Lagrange function must be real-valued, it can also depend on the complex conjugate values  $\psi^*(x, t)$ .

This Lagrange function should be rotation-invariant. There is one more invariance specific for non-relativistic quantum mechanics. Namely, it is known that in quantum mechanics, we can add a constant phase to all the values of  $\psi(x, t)$  without changing the physical meaning. Thus, the Lagrange function should be *phase-invariant*, i.e., invariant with respect to the transformation

$$\psi(x, t) \rightarrow \exp(i \cdot \alpha) \cdot \psi(x, t)$$

for any real constant  $\alpha$ .

In general, a Lagrange function depends both on the fields and on their derivatives. Let us, as usual, denote the time derivative by  $\dot{\psi}$ , and the derivative with respect to  $x_k$  by  $\psi_{,k}$ .

**Definition 1.** By a Lagrange function  $L$  for non-relativistic quantum mechanics, we mean a phase-invariant rotation-invariant real-valued analytical function of the mass  $m$ , its inverse  $m^{-1}$ , fields  $\psi(x, t)$ ,  $\psi^*(x, t)$ , and  $V(x, t)$ , and their derivatives of arbitrary orders with respect to time and spatial coordinates:

$$L(m, m^{-1}, \psi(x, t), \psi_{,k}(x, t), \dot{\psi}(x, t), \dots, \psi^*(x, t), \psi_{,k}^*(x, t), \dot{\psi}^*(x, t), \dots,$$

$$V(x, t), V_k(x, t), \dot{V}(x, t), \dots)$$

### What does scale invariance mean for non-relativistic quantum mechanics?

In relativistic physics, there is a direct connection between units of space and time. In contrast, in the non-relativistic case, there is no such direct connection, so we can independently change the unit for space  $x^i \rightarrow x'^i = \lambda \cdot x^i$  and a unit of time

$$t \rightarrow t' = \mu \cdot t.$$

How do  $L$ ,  $\psi(x, t)$ , and  $V(x, t)$  change under these transformations? A specific feature of quantum measurements is that simple experiments enable us to obtain a unit of action  $\hbar$ ; therefore action  $S = \int L(x, t) dx dt$  must be invariant with respect to scale transformations. Hence,  $L(x, t)$  (which is action/(volume $\times$ time)) must transform as  $L \rightarrow L' = \lambda^{-D} \cdot \mu^{-1} \cdot L$ , where  $D$  is the spatial dimension.

Similarly, since action is energy  $\times$  time, and action is invariant, the potential energy  $V(x, t)$  must transform as  $V \rightarrow V' = \mu^{-1} \cdot V$ .

Energy is mass  $\times$  velocity<sup>2</sup>. We know how energy is transformed and how velocity is transformed. Therefore, for mass, we get  $m \rightarrow m' = \lambda^{-2} \cdot \mu \cdot m$ .

The transformation law for the wave function  $\psi(x, t)$  can be deduced from its physical meaning: the integral  $\int |\psi|^2 dx$  is a probability and is therefore independent (invariant) on the choice of length or time units, i.e. invariant. So,  $|\psi|^2 \sim 1/\text{length}^D$ , hence,  $|\psi|^2 \rightarrow \lambda^{-D} \cdot |\psi|^2$ , and  $\psi \rightarrow \psi' = \lambda^{-D/2} \cdot \psi$ .

If we change the units, then we get the new expression for  $L$

$$L'(x, t) = \lambda^{-D} \cdot \mu^{-1} \cdot L(m, m^{-1}, \psi(x, t), \psi_k(x, t), \dot{\psi}(x, t), \dots, \psi^*(x, t), \psi_k^*(x, t), \dot{\psi}^*(x, t), \dots, V(x, t), V_k(x, t), \dot{V}(x, t), \dots). \quad (1)$$

On the other hand, if we change the units in the original expression, we get

$$L' = L(\lambda^2 \cdot \mu^{-1} \cdot m, \lambda^{-2} \cdot \mu \cdot m^{-1}, \lambda^{-D/2} \cdot \psi, \lambda^{-D/2-1} \cdot \psi_k, \lambda^{-D/2} \cdot \mu \cdot \dot{\psi}, \dots, \lambda^{-D/2} \cdot \psi^*, \lambda^{-D/2-1} \cdot \psi_k^*, \lambda^{-D/2} \cdot \mu \cdot \dot{\psi}^*, \dots, \mu^{-1} \cdot V, \lambda^{-1} \cdot \mu^{-1} \cdot V_k, \mu^{-2} \cdot \dot{V}, \dots). \quad (2)$$

**Definition 2.** We say that a Lagrange function is scale-invariant if for all  $\lambda > 0$  and  $\mu > 0$ , the expressions (1) and (2) coincide.

Now, we are ready to present our main results.

## 3. Main Results

**Theorem 1.** For  $D \geq 3$ , every scale-invariant Lagrange function has the form

$$L = i \cdot b \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{c}{m} \cdot (\nabla \psi \cdot \nabla \psi^*) + d \cdot V \cdot \psi \cdot \psi^* + L_0, \quad (2)$$

where  $b$ ,  $c$ , and  $d$  are real constants, and  $L_0$  is an expression which does not contribute to variational equations.

**Theorem 2.** For  $D = 2$ , every scale-invariant Lagrange function has the form

$$L = i \cdot b \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{c}{m} \cdot (\nabla \psi \cdot \nabla \psi^*) + d \cdot V \cdot \psi \cdot \psi^* + \frac{f}{m} \cdot |\psi|^4 + L_0, \quad (3)$$

where  $b$ ,  $c$ ,  $d$ , and  $f$  are real constants, and  $L_0$  is an expression which does not contribute to variational equations.

**Theorem 3.** For  $D = 1$ , every scale-invariant Lagrange function has the form

$$L = i \cdot b \cdot \left( \psi \cdot \frac{\partial \psi^*}{\partial t} - \psi^* \cdot \frac{\partial \psi}{\partial t} \right) + \frac{c}{m} \cdot (\nabla \psi \cdot \nabla \psi^*) + d \cdot V \cdot \psi \cdot \psi^* + \frac{f}{m} \cdot |\psi|^6 + L_0, \quad (4)$$

where  $b$ ,  $c$ , and  $d$  are real constants, and  $L_0$  is an expression which does not contribute to variational equations.

*Comment.* Thus, we indeed get the desired equations: only Schroedinger for  $D \geq 3$ , Gross-Pitaevskii for  $D = 2$ , and a new nonlinear equation for  $D = 1$ .

### Proof of Theorems 1–3.

**General analysis.** Let us first fix  $m$  and consider only transformations which preserve  $m$ , i.e., transformations for which  $\mu = \lambda^2$ . For these transformations, the formula (1) takes the form

$$\begin{aligned} L'(x, t) = & \lambda^{-(D+2)} \cdot L(m, m^{-1}, \psi(x, t), \psi_{,k}(x, t), \dot{\psi}(x, t), \dots, \\ & \psi^*(x, t), \psi_{,k}^*(x, t), \dot{\psi}^*(x, t), \dots, V(x, t), V_{,k}(x, t), \dot{V}(x, t), \dots), \end{aligned} \quad (5)$$

while the formula (2) takes the form

$$\begin{aligned} L' = & L(m, m^{-1}, \lambda^{-D/2} \cdot \psi, \lambda^{-D/2-1} \cdot \psi_{,k}, \lambda^{-D/2-2} \cdot \dot{\psi}, \dots, \\ & \lambda^{-D/2} \cdot \psi^*, \lambda^{-D/2-1} \cdot \psi_{,k}^*, \lambda^{-D/2-2} \cdot \dot{\psi}^*, \dots, \\ & \lambda^{-2} \cdot V, \lambda^{-3} \cdot V_{,k}, \lambda^{-4} \cdot \dot{V}, \dots). \end{aligned} \quad (6)$$

The expressions (5) and (6) must coincide. Since  $L$  is an analytical function, it is a (possibly infinite) sum of monomials. Since the two analytical functions of  $\lambda^{-1}$  coincide, this means that all the coefficients at the corresponding monomials must coincide.

Each monomial depends on  $\lambda^{-1}$ . All the monomials in the expression (5) multiply by  $\lambda^{-(D+2)}$ . Thus, in the right-hand side, we can only have the monomials which are similarly multiplied. Here:

- $\psi$  is multiplied by  $\lambda^{-D/2}$ ,
- $V$  is multiplied by  $\lambda^{-2}$ ,
- spatial differentiation leads to multiplication by  $\lambda^{-1}$ , and
- temporal differentiation leads to multiplication by  $\lambda^{-1}$ .

Thus, we must have

$$D + 2 = \frac{D}{2} \cdot n_\psi + 2n_V + n_S + 2n_T, \quad (7)$$

where:

- $n_\psi$  is the total number of terms  $\psi$ ,  $\psi^*$ , and their derivatives,
- $n_V$  is the total number of  $V$  and its derivatives,
- $n_S$  is the total number of spatial differentiations, and
- $d_T$  is the total number of differentiations with respect to time.

Terms not depending on  $\psi$  do not affect the action and, thus, do not contribute to the equations; all these terms go directly to  $L_0$ . Thus, we must have  $n_\psi \geq 1$ .

Terms linear (or, in general, of odd order) in  $\psi$  or in its derivatives are not phase-invariant, so we must have  $n_\psi$  even and  $n_\psi \geq 2$ , hence  $n_\psi - 2 \geq 0$ . If we subtract  $D$  from both sides of the equality (5), we conclude that

$$2 = \frac{D}{2} \cdot (n_\psi - 2) + 2n_V + n_S + 2n_T. \quad (8)$$

**Case of  $D \geq 3$ .** For odd  $D \geq 3$ , since the left-hand side is an integer, the difference  $n_\psi - 2$  must be even. If this difference is non-zero, we must thus have  $n_\psi - 2 \geq 2$ . In this case,  $(D/2) \cdot (n_\psi - 2) \geq D \geq 3$ . However, we know that the sum of this product and several non-negative integers is equal to 2. Thus, in this case, we cannot have  $n_\psi - 2 > 0$ , so we must have  $n_\psi - 2 = 0$  and  $n_\psi = 2$ .

Similarly, for even  $D > 2$ , if  $n_\psi - 2 > 0$  then, since  $n_\psi$  is even, we must have

$$n_\psi - 2 \geq 2$$

thus  $(D/2) \cdot (n_\psi - 2) \geq D > 2$ , so we cannot have the sum equal to 2.

Thus, for all  $D \geq 3$ , we must have  $n_\psi = 2$  and so,

$$2 = 2n_V + n_S + 2n_T.$$

Since all three integers  $n_V$ ,  $n_S$ , and  $n_T$  are non-negative, we only have the following three options:

- $n_V = 1$ ,  $n_S = n_T = 0$ ;
- $n_V = 0$ ,  $n_S = 2$ ,  $n_T = 0$ ; and
- $n_V = 0$ ,  $n_S = 0$ ,  $n_T = 1$ .

In all these cases, we have  $n_\psi = 2$ .

In the first case, we get a product of  $V$  and two terms of type  $\psi$  and  $\psi^*$ ; the only way to make it real-valued and phase-invariant is to have  $V \cdot \psi \cdot \psi^*$ . Another possibility would be  $V \cdot (\psi^2 + (\psi^*)^2)$ , but the corresponding term is not phase-invariant.

In the second case, we have two derivatives of two functions  $\psi$ . Due to the requirement that  $L$  is real-valued, one of them must be  $\psi$ , and another one  $\psi^*$ . Due to rotation-invariance, we have two possibilities:  $\psi_{,i} \cdot \psi_{,i}^*$  and  $\psi \cdot \nabla^2 \psi^*$ ; the second term differs from the first one by a full derivative, so we can assume that we get the first term, and add the full derivative to  $L_0$ .

In the third case, we have two functions  $\psi$  and  $\psi^*$  and one time derivative. This leads to the corresponding term in  $L$ .

**Case of  $D = 2$ .** For  $D = 2$ , the above equation takes the form

$$2 = (n_\psi - 2) + 2n_V + n_S + 2n_T.$$

Here, in addition to the case  $n_\psi = 2$ , we can also have the case when  $n_\psi - 2 = 2$  and thus,  $n_\psi = 4$ ; in this case, we have  $n_V = n_S = n_T = 0$ . The only phase-invariant real-valued term of fourth order in  $\psi$  and  $\psi^*$  is  $(\psi \cdot \psi^*)^2 = |\psi|^4$ .

**Case of  $D = 1$ .** For  $D = 1$ , we get

$$2 = \frac{1}{2} \cdot (n_\psi - 2) + 2n_V + n_S + 2n_T.$$

The number of spatial differentiations must be even, otherwise the Lagrange function is not rotation-invariant. Since all the terms in the above equality, except for the term

$$\frac{1}{2} \cdot (n_\psi - 2),$$

are even, this term must also be even. Thus, the only way for it to be non-zero is if this term is  $\geq 2$ . This term cannot be larger than 2 — then we would not be able to have 2 in the left-hand side. Thus, we must have  $(1/2) \cdot (n_\psi - 2) = 2$ , hence  $n_\psi - 2 = 4$  and  $n_\psi = 6$  — and  $n_V = n_S = n_T = 0$ . Similarly to the case  $D = 2$ , the only phase-invariant real-valued term of sixth order in  $\psi$  and  $\psi^*$  is the term  $(\psi \cdot \psi^*)^3 = |\psi|^6$ .

**Final part of the proof.** We have almost proved the theorems, except for the dependence on  $m$ . To finalize the proof, we can take the expression that we have obtained so far,

- explicitly mention that all the coefficients  $a, b, \dots$  should depend on  $m$ , and
- describe the requirement that the resulting formula must be invariant with respect to the scaling transformation corresponding to all possible pairs  $(\lambda, \mu)$ .

This enables us to find the exact dependence of all the coefficients on  $m$ .

The theorems are proven.

## Acknowledgments

This work was supported in part by the US National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721.

## REFERENCES

1. Feynman R.P., Leighton R.B., Sands M.L. The Feynman Lectures on Physics. Addison-Wesley, Redwood City, California, 2005.

2. Finkelstein A.M., Kreinovich V. Derivation of Einstein's, Brans-Dicke and other equations from group considerations // On Relativity Theory. Proceedings of the Sir Arthur Eddington Centenary Symposium, Nagpur India 1984. Vol. 2. Y. Choque-Bruhat and T.M. Karade (eds), World Scientific, Singapore, 1985. P. 138–146.
3. Finkelstein A.M., Kreinovich V., Zapatin R.R. Fundamental physical equations uniquely determined by their symmetry groups // Lecture Notes in Mathematics. Springer-Verlag, Berlin-Heidelberg-N.Y., 1986. Vol. 1214. P. 159–170.
4. Green M., Schwarz J.H., Witten E. Superstring Theory. Vols. 1 and 2. Cambridge University Press, Cambridge, Massachusetts, 1987.
5. Grossm E.P. Structure of a quantized vortex in boson systems // Il Nuovo Cimento. 1961. Vol. 20, No. 3. P. 545–457.
6. Group theory in physics: proceedings of the international symposium held in honor of Prof. Marcos Moshinsky. Cocoyoc, Morelos, Mexico, 1991. American Institute of Physics, N.Y., 1992.
7. Kreinovich V. Derivation of the Schroedinger equations from scale invariance // Theoretical and Mathematical Physics. 1976. Vol. 8, No. 3. P. 282–285.
8. Kreinovich V., Liu G. We live in the best of possible worlds: Leibniz's insight helps to derive equations of modern physics / Pisano R., Fichant M., Bussotti P. Oliveira A.R.E. (eds.). Leibniz and the Dialogue between Sciences, Philosophy and Engineering, 1646-2016, New Historical and Epistemological Insights. The College Publications, London, 2017. P. 207–226.
9. Pitaevskii L.P. Vortex lines in an imperfect Bose gas // Soviet Physics JETP. 1961. Vol. 13, No. 2. P. 451–454.
10. Pitaevskii L.P., Stringari S. Bose-Einstein Condensation. Claredon Press, Oxford, UK, 2003.

### **ВЫВОД НЕЛИНЕЙНОГО УРАВНЕНИЯ ШРЁДИНГЕРА В ВАРИАНТЕ ГРОССА-ПИТАЕВСКОГО ИЗ МАСШТАБНОЙ ИНВАРИАНТНОСТИ**

**О. Кошелева**

к.ф.-м.н., доцент, e-mail: olgak@utep.edu

**В. Крейнович**

к.ф.-м.н., профессор, e-mail: vladik@utep.edu

Техасский университет в Эль Пасо, США

**Аннотация.** Известно, что в обычном трёхмерном пространстве уравнение Шрёдингера может быть получено из масштабной инвариантности. Ввиду того, что, согласно современной физике, фактическая размерность собственного пространства может отличаться от 3, желательно проанализировать, что происходит в других пространственных измерениях  $D$ . Оказывается, что хотя для  $D > 3$  мы получаем только уравнение Шрёдингера, для  $D = 2$  мы получаем также нелинейное уравнение Шрёдингера в варианте Гросса-Питаевского, описывающее квантовую систему идентичных бозонов, и для  $D = 1$  мы также получаем новую нелинейную версию уравнения Шрёдингера.

**Ключевые слова:** масштабная инвариантность, нелинейное уравнение Шрёдингера, уравнение Гросса-Питаевского, система идентичных бозонов.

*Дата поступления в редакцию: 21.03.2017*

## WHY LINEAR INTERPOLATION?

**Andrzej Pownuk**

Ph.D. (Phys.-Math.), Instructor, e-mail: ampownuk@utep.edu

**Vladik Kreinovich**

Ph.D. (Phys.-Math.), Professor, e-mail: vladik@utep.edu

University of Texas at El Paso, El Paso, Texas 79968, USA

**Abstract.** Linear interpolation is the computationally simplest of all possible interpolation techniques. Interestingly, it works reasonably well in many practical situations, even in situations when the corresponding computational models are rather complex. In this paper, we explain this empirical fact by showing that linear interpolation is the only interpolation procedure that satisfies several reasonable properties such as consistency and scale-invariance.

**Keywords:** linear interpolation, scale-invariance.

### 1. Formulation of the Problem

**Need for interpolation.** In many practical situations, we know that the value of a quantity  $y$  is uniquely determined by the value of some other quantity  $x$ , but we do not know the exact form of the corresponding dependence  $y = f(x)$ .

To find this dependence, we measure the values of  $x$  and  $y$  in different situations. As a result, we get the values  $y_i = f(x_i)$  of the unknown function  $f(x)$  for several values  $x_1, \dots, x_n$ . Based on this information, we would like to predict the value  $f(x)$  for all other values  $x$ . When  $x$  is between the smallest and the largest of the values  $x_i$ , this prediction is known as the *interpolation*; for values  $x$  smaller than the smallest of  $x_i$  or larger than the largest of  $x_i$ , this prediction is known as *extrapolation*; see, e.g., [1].

**Simplest possible case of interpolation.** The simplest possible case of interpolation is when we only know the values  $y_1 = f(x_1)$  and  $y_2 = f(x_2)$  of the function  $f(x)$  at two points  $x_1 < x_2$ , and we would like to predict the value  $f(x)$  at points  $x \in (x_1, x_2)$ .

**In many cases, linear interpolations works well: why?** One of the most well-known interpolation techniques is based on the assumption that the function  $f(x)$  is linear on the interval  $[x_1, x_2]$ . Under this assumption, we get the following formula for  $f(x)$ :

$$f(x) = \frac{x - x_1}{x_2 - x_1} \cdot f(x_2) + \frac{x_2 - x}{x_2 - x_1} \cdot f(x_1).$$

This formula is known as *linear interpolation*.



The usual motivation for linear interpolation is simplicity: linear functions are the easiest to compute, and this explains why we use linear interpolation.

An interesting empirical fact is that in many practical situations, linear interpolation works reasonably well. We know that in computational science, often very complex computations are needed, so we cannot claim that nature prefers simple functions. There should be another reason for the empirical fact that linear interpolation often works well.

**What we do.** In this paper, we show that linear interpolation can indeed be derived from fundamental principles.

## 2. Analysis of the Problem: What Are Reasonable Properties of an Interpolation

**What is interpolation.** We want to be able, given values  $y_1$  and  $y_2$  of the unknown function at points  $x_1$  and  $x_2$ , and a point  $x \in (x_1, x_2)$ , to provide an estimate for  $f(x)$ . In other words, we need a function that, given the values  $x_1, y_1, x_2, y_2$ , and  $x$ , generates the estimate for  $f(x)$ . We will denote this function by  $I(x_1, y_1, x_2, y_2, x)$ .

What are the reasonable properties of this function?

**Conservativeness.** If both observed values  $y_i = f(x_i)$  are smaller than or equal to some threshold value  $y$ , it is reasonable to expect that all intermediate values of  $f(x)$  should also be smaller than or equal to  $y$ . Thus, if  $y_1 \leq y$  and  $y_2 \leq y$ , then we should have  $I(x_1, y_1, x_2, y_2, x) \leq y$ .

In particular, for  $y = \max(y_1, y_2)$ , we conclude that

$$I(x_1, y_1, x_2, y_2, x) \leq \max(y_1, y_2).$$

Similarly, if both observed values  $y_i = f(x_i)$  are greater than or equal to some threshold value  $y$ , it is reasonable to expect that all intermediate values of  $f(x)$  should also be greater than or equal to  $y$ . Thus, if  $y \leq y_1$  and  $y \leq y_2$ , then we should have  $y \leq I(x_1, y_1, x_2, y_2, x)$ .

In particular, for  $y = \min(y_1, y_2)$ , we conclude that

$$\min(y_1, y_2) \leq I(x_1, y_1, x_2, y_2, x)$$

These two requirements can be combined into a single double inequality

$$\min(y_1, y_2) \leq I(x_1, y_1, x_2, y_2, x) \leq \max(y_1, y_2).$$

We will call this property *conservativeness*.

**$x$ -scale-invariance.** The numerical value of a physical quantity depends on the choice of the measuring unit and on the starting point. If we change the starting point to the one which is  $b$  units smaller, then  $b$  is added to all the numerical values. Similarly, if we replace a measuring unit by a one which is  $a > 0$  times smaller, then all the numerical values are multiplied by  $a$ . If we perform both changes, then each original value  $x$  is replaced by the new value  $x' = a \cdot x + b$ .

For example, if we know the temperature  $x$  in Celsius, then the temperature  $x'$  in Fahrenheit can be obtained as  $x' = 1.8 \cdot x + 32$ .

It is reasonable to require that the interpolation procedure should not change if we simply change the measuring unit and the starting point — without changing the actual physical quantities. In other words, it is reasonable to require that

$$I(a \cdot x_1 + b, y_1, a \cdot x_2 + b, y_2, a \cdot x + b) = I(x_1, y_1, x_2, y_2, x).$$

**$y$ -scale-invariance.** Similarly, we can consider different units for  $y$ . The interpolation result should not change if we simply change the starting point and the measuring unit. So, if we replace  $y_1$  with  $a \cdot y_1 + b$  and  $y_2$  with  $a \cdot y_2 + b$ , then the result of interpolation should be obtained by a similar transformation from the previous result:  $I \rightarrow a \cdot I + b$ . Thus, we require that

$$I(x_1, a \cdot y_1 + b, x_2, a \cdot y_2 + b, x) = a \cdot I(x_1, y_1, x_2, y_2, x) + b.$$

**Consistency.** Let us assume that we have  $x_1 \leq x'_1 \leq x \leq x'_2 \leq x_2$ . Then, the value  $f(x)$  can be estimated in two different ways:

- we can interpolate directly from the values  $y_1 = f(x_1)$  and  $y_2 = f(x_2)$ , getting  $I(x_1, y_1, x_2, y_2, x)$ , or
- we can first use interpolation to estimate the values  $f(x'_1) = I(x_1, y_1, x_2, y_2, x'_1)$  and  $f(x'_2) = I(x_1, y_1, x_2, y_2, x'_2)$ , and then use these two estimates to estimate  $f(x)$  as

$$\begin{aligned} I(x_1, f(x'_1), x_2, f(x'_2), x) &= \\ &= I(x'_1, I(x_1, y_1, x_2, y_2, x'_1), x'_2, I(x_1, y_1, x_2, y_2, x'_2), x). \end{aligned}$$

It is reasonable to require that these two ways lead to the same estimate for  $f(x)$ :

$$I(x_1, y_1, x_2, y_2, x) = I(x'_1, I(x_1, y_1, x_2, y_2, x'_1), x'_2, I(x_1, y_1, x_2, y_2, x'_2), x).$$

**Continuity.** Most physical dependencies are continuous. Thus, when the two value  $x$  and  $x'$  are close, we expect the estimates for  $f(x)$  and  $f(x')$  to be also close. Thus, it is reasonable to require that the interpolation function  $I(x_1, y_1, x_2, y_2, x)$  is continuous in  $x$  — and that for both  $i = 1, 2$  the value  $I(x_1, y_1, x_2, y_2, x)$  converges to  $f(x_i)$  when  $x \rightarrow x_i$ .

Now, we are ready to formulate our main result.

### 3. Main Result

**Definition 1.** By an interpolation function, we mean a function  $I(x_1, y_1, x_2, y_2, x)$  which is defined for all  $x_1 < x < x_2$  and which has the following properties:

- conservativeness:

$$\min(y_1, y_2) \leq I(x_1, y_1, x_2, y_2, x) \leq \max(y_1, y_2)$$

for all  $x_i, y_i$ , and  $x$ ;

- *x-scale-invariance*:  $I(a \cdot x_1 + b, y_1, a \cdot x_2 + b, y_2, a \cdot x + b) = I(x_1, y_1, x_2, y_2, x)$  for all  $x_i, y_i, x, a > 0$ , and  $b$ ;
- *y-scale invariance*:  $I(x_1, a \cdot y_1 + b, x_2, a \cdot y_2 + b, x) = a \cdot I(x_1, y_1, x_2, y_2, x) + b$  for all  $x_i, y_i, x, a > 0$ , and  $b$ ;
- consistency:

$$I(x_1, y_1, x_2, y_2, x) = I(x'_1, I(x_1, y_1, x_2, y_2, x'_1), x'_2, I(x_1, y_1, x_2, y_2, x'_2), x)$$

for all  $x_i, x'_i, y_i$ , and  $x$ ; and

- continuity: the expression  $I(x_1, y_1, x_2, y_2, x)$  is a continuous function of  $x$ ,  $I(x_1, y_1, x_2, y_2, x) \rightarrow y_1$  when  $x \rightarrow x_1$  and  $I(x_1, y_1, x_2, y_2, x) \rightarrow y_2$  when  $x \rightarrow x_2$ .

**Proposition.** The only interpolation function satisfying all the properties from Definition 1 is the linear interpolation

$$I(x_1, y_1, x_2, y_2, x) = \frac{x - x_1}{x_2 - x_1} \cdot y_2 + \frac{x_2 - x}{x_2 - x_1} \cdot y_1. \quad (1)$$

**Discussion.** Thus, we have indeed explained that linear interpolation follows from the fundamental principles – which may explain its practical efficiency.

**Proof.**

1°. When  $y_1 = y_2$ , the conservativeness property implies that  $I(x_1, y_1, x_2, y_1, x) = y_1$ . Thus, to complete the proof, it is sufficient to consider two remaining cases: when  $y_1 < y_2$  and when  $y_2 < y_1$ .

We will consider the case when  $y_1 < y_2$ . The case when  $y_2 < y_1$  is considered similarly. So, in the following text, without losing generality, we assume that  $y_1 < y_2$ .

2°. When  $y_1 < y_2$ , then we can get these two values  $y_1$  and  $y_2$  as  $y_1 = a \cdot 0 + b$  and  $y_2 = a \cdot 1 + b$  for  $a = y_2 - y_1$  and  $b = y_1$ . Thus, the *y-scale-invariance* implies that

$$I(x_1, y_1, x_2, y_2, x) = (y_2 - y_1) \cdot I(x_1, 0, x_2, 1, x) + y_1. \quad (2)$$

If we denote  $J(x_1, x_2, x) \stackrel{\text{def}}{=} I(x_1, 0, x_2, 1, x)$ , then we get

$$\begin{aligned} I(x_1, y_1, x_2, y_2, x) &= (y_2 - y_1) \cdot J(x_1, x_2, x) + y_1 = \\ &= J(x_1, x_2, x) \cdot y_2 + (1 - J(x_1, x_2, x)) \cdot y_1. \end{aligned} \quad (3)$$

3°. Since  $x_1 < x_2$ , we can similarly get these two values  $x_1$  and  $x_2$  as  $x_1 = a \cdot 0 + b$  and  $x_2 = a \cdot 1 + b$ , for  $a = x_2 - x_1$  and  $b = x_1$ . Here,  $x = a \cdot r + b$ , where

$$r = \frac{x - b}{a} = \frac{x - x_1}{x_2 - x_1}.$$

Thus, the  $x$ -scale invariance implies that

$$J(x_1, x_2, x) = J\left(0, 1, \frac{x - x_1}{x_2 - x_1}\right).$$

So, if we denote  $w(r) \stackrel{\text{def}}{=} J(0, 1, r)$ , we then conclude that

$$J(x_1, x_2, x) = w\left(\frac{x - x_1}{x_2 - x_1}\right),$$

and thus, the above expression (3) for  $I(x_1, y_1, x_2, y_2, x)$  in terms of  $J(x_1, x_2, x)$  takes the following simplified form:

$$I(x_1, y_1, x_2, y_2, x) = w\left(\frac{x - x_1}{x_2 - x_1}\right) \cdot y_2 + \left(1 - w\left(\frac{x - x_1}{x_2 - x_1}\right)\right) \cdot y_1. \quad (4)$$

To complete our proof, we need to show that  $w(r) = r$  for all  $r \in (0, 1)$ .

4°. Let us now use consistency.

Let us take  $x_1 = y_1 = 0$  and  $x_2 = y_2 = 1$ , then

$$I(0, 0, 1, 1, x) = w(x) \cdot 1 + (1 - w(x)) \cdot 0 = w(x).$$

Let us denote  $\alpha \stackrel{\text{def}}{=} w(0.5)$ .

By consistency, for  $x = 0.25 = \frac{0 + 0.5}{2}$ , the value  $w(0.25)$  can be obtained if we apply the same interpolation procedure to  $w(0) = 0$  and to  $w(0.5) = \alpha$ . Thus, we get

$$w(0.25) = \alpha \cdot w(0.5) + (1 - \alpha) \cdot w(0) = \alpha^2.$$

Similarly, for  $x = 0.75 = \frac{0.5 + 1}{2}$ , the value  $w(0.75)$  can be obtained if we apply the same interpolation procedure to  $w(0.5) = \alpha$  and to  $w(1) = 1$ . Thus, we get

$$w(0.75) = \alpha \cdot w(1) + (1 - \alpha) \cdot w(0.5) = \alpha \cdot 1 + (1 - \alpha) \cdot \alpha = 2\alpha - \alpha^2.$$

Finally, for  $x = 0.5 = \frac{0.25 + 0.75}{2}$ , the value  $w(0.5)$  can be obtained if we apply the same interpolation procedure to  $w(0.25) = \alpha^2$  and to  $w(0.75) = 2\alpha - \alpha^2$ . Thus, we get

$$\begin{aligned} w(0.5) &= \alpha \cdot w(0.75) + (1 - \alpha) \cdot w(0.25) = \\ &= \alpha \cdot (2\alpha - \alpha^2) + (1 - \alpha) \cdot \alpha^2 = 3\alpha^2 - 2\alpha^3. \end{aligned}$$

By consistency, this estimate should be equal to our original estimate  $w(0.5) = \alpha$ , i.e., we must have

$$3\alpha^2 - 2\alpha^3 = \alpha. \quad (5)$$

5°. One possible solution is to have  $\alpha = 0$ . In this case, we have  $w(0.5) = 0$ . Then, we have

$$w(0.75) = \alpha \cdot w(1) + (1 - \alpha) \cdot w(0.5) = 0,$$

and by induction, we can show that in this case,  $w(1 - 2^{-n}) = 0$  for each  $n$ . In this case,  $1 - 2^{-n} \rightarrow 1$ , but  $w(1 - 2^{-n}) \rightarrow 0$ , which contradicts to the continuity requirement, according to which  $w(1 - 2^{-n}) \rightarrow w(1) = 1$ .

Thus, the value  $\alpha = 0$  is impossible, so  $\alpha \neq 0$ , and we can divide both sides of the above equality (5) by  $\alpha$ .

As a result, we get a quadratic equation

$$3\alpha - 2\alpha^2 = 1,$$

which has two solutions:  $\alpha = 1$  and  $\alpha = 0.5$ .

6°. When  $\alpha = 1$ , we have  $w(0.5) = 1$ . Then, we have

$$w(0.25) = \alpha \cdot w(0.5) + (1 - \alpha) \cdot w(0) = 1,$$

and by induction, we can show that in this case,  $w(2^{-n}) = 1$  for each  $n$ . In this case,  $2^{-n} \rightarrow 0$ , but  $w(2^{-n}) \rightarrow 1$ , which contradicts to the continuity requirement, according to which  $w(2^{-n}) \rightarrow w(0) = 0$ .

Thus, the value  $\alpha = 1$  is impossible, so  $\alpha = 0.5$ .

7°. For  $\alpha = 0.5$ , we have  $w(0) = 0$ ,  $w(0.5) = 0.5$ , and  $w(1) = 1$ . Let us prove, by induction over  $q$ , that for every binary-rational number  $r = \frac{p}{2^q} \in [0, 1]$ , we have  $w(r) = r$ .

Indeed, the base case  $q = 1$  is proven. Let us assume that we have proven it for  $q - 1$ , let us prove it for  $q$ . If  $p$  is even, i.e., if  $p = 2k$ , then  $\frac{2k}{2^q} = \frac{k}{2^{q-1}}$ , so the desired equality comes from the induction assumption. If  $p = 2k + 1$ , then

$$r = \frac{p}{2^q} = \frac{2k + 1}{2^q} = 0.5 \cdot \frac{2k}{2^q} + 0.5 \cdot \frac{2 \cdot (k + 1)}{2^q} = 0.5 \cdot \frac{k}{2^{q-1}} + 0.5 \cdot \frac{k + 1}{2^{q-1}}.$$

By consistency, we thus have

$$w(r) = 0.5 \cdot w\left(\frac{k}{2^{q-1}}\right) + 0.5 \cdot w\left(\frac{k + 1}{2^{q-1}}\right).$$

By induction assumption, we have

$$w\left(\frac{k}{2^{q-1}}\right) = \frac{k}{2^{q-1}} \text{ and } w\left(\frac{k + 1}{2^{q-1}}\right) = \frac{k + 1}{2^{q-1}}.$$

So, the above formula takes the form

$$w(r) = 0.5 \cdot \frac{k}{2^{q-1}} + 0.5 \cdot \frac{k+1}{2^{q-1}},$$

hence  $w(r) = \frac{2k+1}{2^q} = r$ .

The statement is proven.

8°. The equality  $w(r) = r$  is true for all binary-rational numbers. Any real number  $x$  from the interval  $[0, 1]$  is a limit of such numbers — namely, truncates of its infinite binary expansion. Thus, by continuity, we have  $w(x) = x$  for all  $x$ .

Substituting  $w(x) = x$  into the above formula (4) for  $I(x_1, y_1, x_2, y_2, x)$  leads exactly to linear interpolation.

The proposition is proven.

## Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award “UTEP and Prudential Actuarial Science Academy and Pipeline Initiative” from Prudential Foundation.

## REFERENCES

1. Burden R.L., Faires J.D., Burden A.M. Numerical Analysis. Cengage Learning, Boston, Massachusetts, 2015.

## ПОЧЕМУ ЛИНЕЙНАЯ ИНТЕРПОЛЯЦИЯ?

**А. Повнук**

к.ф.-м.н., ст. преподаватель, e-mail: ampownuk@utep.edu

**В. Крейнович**

к.ф.-м.н., профессор, e-mail: vladik@utep.edu

Техасский университет в Эль Пасо, США

**Аннотация.** Линейная интерполяция — это простейший в вычислительном отношении из всех возможных методов интерполяции. Интересно, что он работает достаточно хорошо во многих практических ситуациях, даже в ситуациях, когда соответствующие вычислительные модели довольно сложны. В этой статье мы объясняем этот эмпирический факт, показывая, что линейная интерполяция является единственной процедурой интерполяции, которая удовлетворяет нескольким разумным свойствам, таким как согласованность и масштабная инвариантность.

**Ключевые слова:** линейная интерполяция, масштабная инвариантность.

*Дата поступления в редакцию: 09.04.2017*

## ПОСТРОЕНИЕ ЭФФЕКТИВНЫХ СЕНСОРНЫХ СЕТЕЙ С УЧЁТОМ СТОИМОСТНЫХ ЗАТРАТ

**С.Н. Астраков**<sup>1,2</sup>

доцент, к.ф.-м.н., e-mail: astrakov90@gmail.com

**А.Г. Квашнин**<sup>1</sup>

к.ф.-м.н., e-mail: kvashninag@kti.sbras.ru

**Л.А. Короленко**<sup>1</sup>

инженер-программист, e-mail: lakorolenko@gmail.com

<sup>1</sup>Институт вычислительных технологий СО РАН

<sup>2</sup>Новосибирский исследовательский национальный университет

**Аннотация.** В работе рассматриваются способы проектирования беспроводных сенсорных сетей при помощи моделей регулярных круговых покрытий. Ставится задача построения наиболее эффективной сенсорной сети при заданной сложности её структуры. При этом учитывается стоимость типового сенсора и затраты на эксплуатацию сети. Размер зоны действия сенсорных устройств соответствует радиусам кругов. Сложность структуры сенсорной сети определяется количеством различных размеров кругов (одного, двух и более типов) и способом их расположения. В связи с этим, предложены принципы классификации регулярных покрытий, основанные на понятии минимального фрагмента. Представлены методы расчёта оптимального количества сенсорных устройств для ограниченных областей при заданной структуре регулярной сенсорной сети.

**Ключевые слова:** сенсорные сети, круговые модели покрытий, плотность покрытия, регулярная структура покрытия, оптимизация затрат.

### Введение

Круговые покрытия пространственных областей являются основными моделями при проектировании беспроводных сенсорных сетей. Предположим, что зоной действия сенсора является круг. Тогда покрытие области  $S$  множеством кругов  $M$  является моделью беспроводной сенсорной сети. Каждая точка области принадлежит, по крайней мере, одному кругу покрытия. Плотность покрытия определяется отношением суммарной площади кругов множества  $M$  к площади области  $S$ . Плотность покрытия и способ расположения кругов определяют эффективность сенсорной сети. Обычно рассматриваются регулярные покрытия плоскости, структура которых локально устроена одинаковым образом. Если все круги имеют одинаковый размер, то такие покрытия задаются при помощи решёток [1]. Регулярные покрытия с различными размерами кругов

(двух и более типов) устроены более сложно (рис. 1). Плотность таких покрытий зависит не только от расположения центров кругов, но и от соотношения между радиусами [2–4]. Наиболее естественным способом описания структуры регулярных покрытий является разбиение плоскости на типовые фрагменты (плитки) с одинаковым покрытием. В этом случае плотность покрытия будет определяться плотностью покрытия одного фрагмента. Это позволяет не только строго определять характеристики покрытий, но и строить новые модели сенсорных сетей.

Отметим, что задачи о покрытии плоскости кругами «в целом» отличаются от задач покрытия ограниченных областей. Если рассматривать покрытия ограниченных областей, то наличие границ существенно усложняет точные расчёты и процесс оптимизации. В этом случае основной целью является поиск оптимального числа сенсоров, необходимых для обслуживания области. В [5] приведены алгоритмы расчётов оптимального числа сенсоров, обеспечивающих не только контроль области, но и передачу данных в сети.

При оценке количества кругов, требуемых для покрытия, можно учитывать не только площадь области, но и протяжённость границы. Так в работе [6] показано, что число  $n = \lceil 2a/\sqrt{27} + 2p/(\pi\sqrt{3}) + 1 \rceil$  единичных кругов является достаточным для покрытия любой простой связной области с площадью  $a$  и периметром  $p$ . Заметим, что в данном случае круги имеют фиксированный размер, а уменьшение количества кругов достигается за счёт малой плотности покрытия.

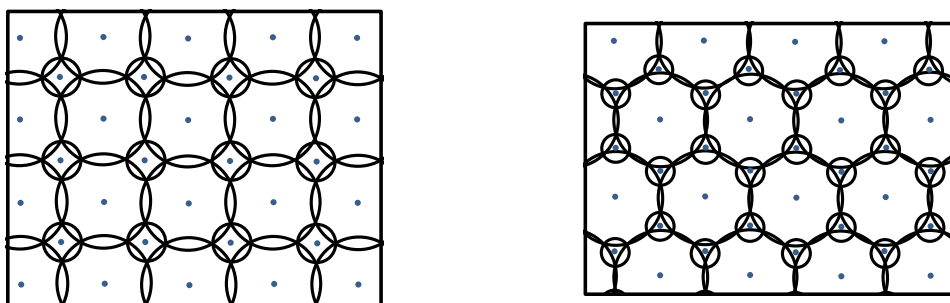


Рис. 1. Примеры регулярных покрытий плоскости кругами двух видов

При покрытии областей с границами невозможно избежать выхода кругов за границу области, но можно минимизировать возникающие потери. В [7, 8] показано, как можно строго учесть граничные эффекты для регулярных покрытий бесконечной полосы (рис. 2). В этих работах рассмотрены наиболее интересные модели покрытий кругами одного, двух и трёх видов. Для этих моделей найдены оптимальные параметры (расположение кругов и точные соотношения между радиусами), которые определяют минимальную плотность покрытия.



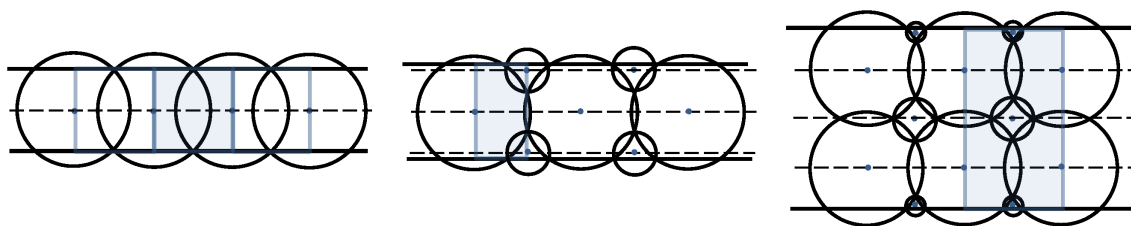


Рис. 2. Примеры регулярных покрытий полосы кругами одного, двух и трёх видов

В общем случае для ограниченных (компактных) плоских областей возможны два основных направления исследования: (1) рассматривать регулярные покрытия, приближённо учитывая искажения в приграничной части или (2) оптимизировать покрытие области при заданном количестве кругов, отказавшись от регулярности. В данной работе мы будем рассматривать задачи только первого направления для ограниченных областей заданной площади. Будем предполагать, что известна стоимость типовых сенсорных устройств, а радиус действия сенсора является регулируемым параметром. Понятно, что радиус действия сенсора соответствует мощности устройства, а мощность определяет энергетические затраты и, соответственно, эксплуатационные затраты. Такой подход позволяет достаточно точно определить количество сенсорных устройств, необходимых для экономичного мониторинга ограниченной области. Важно заметить, что принципиальная идея метода состоит в том, что пропорциональное изменение размера кругов не меняет плотность покрытия, но позволяет минимизировать стоимость и энергетические затраты. Для решения практических задач методика расчёта допускает учёт формы области и протяжённость её границы.

Настоящая работа построена следующим образом. В разделе 1 введена классификации покрытий, позволяющая однозначно определять уровень сложности структуры покрытия. Способ построения функции затрат рассмотрен в разделе 2. В разделе 3 приведены примеры оптимизации сенсорных сетей, имеющих заданные структуры с одним и двумя типами устройств. В разделе 4 приведены результаты вычислительных экспериментов. В разделе 5 дан анализ полученных результатов и определены перспективы дальнейших исследований.

## 1. Классификация регулярных покрытий

Для дальнейших исследований нам необходимо ввести классификацию регулярных покрытий, которая позволяет описывать структуру покрытия и определять уровень его сложности. В нашем случае структура покрытия — это способ взаимного расположения соседних кругов и соотношение между их размерами. Предложенные принципы классификации позволяют однозначно определять уровень сложности покрытия при помощи выделения минимального фрагмента.

Очевидно, что плотность покрытия всегда больше единицы. При оптимизации структуры покрытия стараются уменьшить плотность, так как это приводит

к уменьшению энергетических затрат на функционирование сенсорной сети. С другой стороны, оптимизация проводится при фиксированной сложности, так как это является существенным преимуществом проектирования и построения сенсорной сети.

Введём следующее обозначение для различных классов регулярных покрытий:

$$Cov(F : k_1/p_1, k_2/p_2, \dots, k_n/p_n),$$

где  $n$  — число различных типов кругов;  $k_i$  — число кругов вида  $i$ , участвующих в покрытии типового фрагмента  $F$ ;  $p_i$  — доли кругов вида  $i$ , покрывающих фрагмент  $F$ .

Далее будем считать, что фрагмент  $F$  является минимальным, то есть его нельзя разбить на части, которые тоже можно считать фрагментами. В этом случае обозначение позволяет однозначно восстановить геометрическую структуру покрытия. Кроме того, если фрагмент является минимальным, то значение  $n$  можно считать основным показателем уровня сложности покрытия  $Ln$ . Например, для первого покрытия с уровнем сложности  $L2$ , изображённого на рисунке 1, минимальным фрагментом будет прямоугольный треугольник с равными катетами (рис. 3). Обозначение покрытия будет иметь вид:  $Cov(F : 1_8, 1_8)$ , так как круги, покрывающие фрагмент, участвуют в его покрытии своими восьмыми частями. Заметим, что знак дроби в обозначении будем опускать, чтобы запись была более удобной.

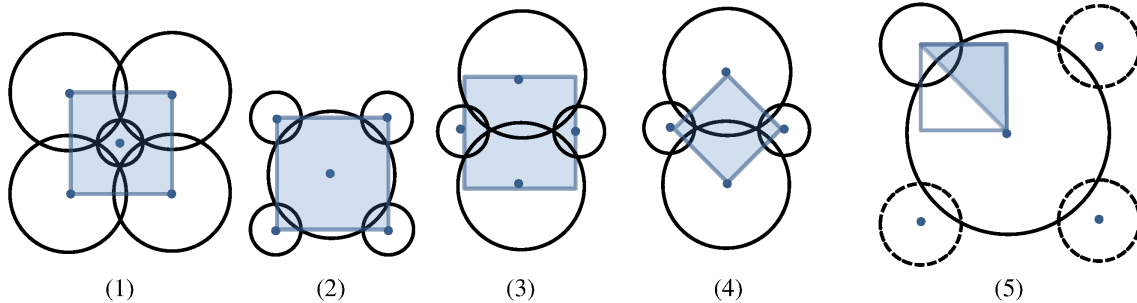


Рис. 3. Фрагменты (1)-(4) и минимальный фрагмент (5) покрытия  $Cov(F : 1_8, 1_8)$

Интересно сравнить соответствующие обозначения для других фрагментов данного покрытия, изображённых на рисунке 3:

$$Cov(F_1 : 4_4, 1_1), Cov(F_2 : 1_1, 4_4), Cov(F_3 : 2_2, 2_2), Cov(F_4 : 2_4, 2_4).$$

Согласованность этих обозначений состоит в том, что соотношение между дробями остаётся одинаковым:  $(k_1/p_1) : (k_2/p_2) = 1 : 1$ . Это значит, что в среднем на один большой круг покрытия приходится один маленький.

Заметим, что класс покрытий с заданным фрагментом и фиксированным уровнем сложности имеет конечное число моделей с различными структурами. Структура покрытия определяется положением кругов относительно данного фрагмента, точнее положением центра круга. Если фрагмент является

$k$ –угольником, то возможными положениями центра круга являются либо вершина, либо сторона, либо внутренняя часть. Таким образом, можно рассмотреть все комбинации для положения  $n$  центров кругов в типовом фрагменте, исследовать соответствующие структуры и определить оптимальное покрытие в заданном классе.

## 2. Построение типовой функции затрат на обслуживание сети

При построении типовой функции затрат для сенсорной сети будем учитывать следующие характеристики сенсорных датчиков: стоимость устройств и энергию потребления за период эксплуатации. Для ограниченных областей можно заранее выбрать эффективную структуру покрытия заданного класса и рассчитать характеристики датчиков так, чтобы затраты были минимальными.

### 2.1. Функции затрат для покрытий сложности $L1$

Для области площади  $S$  затраты можно определить через радиусы кругов  $R$  и их количество  $N$ . Предполагаем, что все затраты выражаются в стоимостной форме. Учитывая особенности отдельных затрат, выделим три вида:

- $E_0$  — разовые затраты, например, стоимость проектирования,
- $E_1$  — стоимость устройств и их установки,
- $E_2$  — затраты на работу сети за период эксплуатации.

Пусть стоимость и установка одного комплекта оборудования определяется величиной  $P$ , а эксплуатационные затраты на работу одного комплекта равны  $Q$ . Тогда функция затрат имеет вид:

$$E = E(N, P, Q) = E_0 + E_1(N, P) + E_2(N, Q).$$

Считаем, что  $E_1(N, P) = NP$  и  $E_2(N, Q) = NQ$ . Плотность покрытия по определению равна  $D = \frac{N\pi R^2}{S}$ , поэтому  $N = \frac{SD}{\pi R^2}$ . Следовательно, общие затраты можно выразить через величину  $R$ :

$$E = E_0 + NP + NQ = E_0 + \frac{SD}{\pi R^2}(P + Q). \quad (1)$$

Показатель эксплуатационных затрат  $Q$  зависит от мощности устройства (дальности действия сигнала), поэтому с точностью до коэффициента  $\mu$  он определяется через радиус круга:  $Q = \mu R^\alpha$ . Принято считать [1], что  $\alpha \in (2, 6)$ .

Учитывая (1), стоимостные затраты на проектирование и эксплуатацию сенсорной сети определяются следующей функцией:

$$E = E(R) = E_0 + \frac{SD}{\pi R^2}(P + \mu R^\alpha). \quad (2)$$

**Теорема 1.** Пусть для мониторинга (покрытия) области с площадью  $S$  используется регулярная структура покрытия плотности  $D$ , а затраты на

установку и эксплуатацию одного сенсорного устройства равны, соответственно,  $P$  и  $Q = \mu R^\alpha$ ,  $\alpha \in (2, 6)$ . Тогда приближенная оценка оптимального радиуса действия сенсорного устройства и общего количества устройств определяется следующими соотношениями:

$$R = \left( \frac{2P}{\mu(\alpha - 2)} \right)^{1/\alpha}, \quad N \approx \frac{SD}{\pi} \left( \frac{\mu(\alpha - 2)}{2P} \right)^{2/\alpha}.$$

*Доказательство.* Исследуем функцию (2) на экстремум:

$$\frac{dE}{dR} = \frac{SD}{\pi} \left( P \frac{(-2)}{R^3} + \mu(\alpha - 2)R^{\alpha-3} \right).$$

Из этого соотношения определяем условие на минимум  $\mu(\alpha - 2)R^\alpha = 2P$  и оптимальное значения радиуса  $R = \left( \frac{2P}{\mu(\alpha-2)} \right)^{1/\alpha}$ .

Учитывая, что число  $N$  целое, находим его приближенное значение:

$$N = \frac{SD}{\pi R^2} \approx \frac{SD}{\pi} \left( \frac{\mu(\alpha - 2)}{2P} \right)^{2/\alpha}. \quad (3)$$

■

**Следствие 1.** Используя результаты теоремы 1, можно определить минимальные затраты на проектирование и эксплуатацию сенсорной сети по формуле (2)

$$E_{min} = E_0 + \frac{SDP\alpha}{\pi(\alpha - 2)} \left( \frac{\mu(\alpha - 2)}{2P} \right)^{2/\alpha}.$$

**Замечание 1.** Если показатель  $\alpha$  будет достаточно близким к 2, то значение радиуса будет сколь угодно большим. В этом случае для мониторинга области достаточно брать только одно сенсорное устройство.

**Замечание 2.** Формула (3) показывает, что число сенсоров  $N$  пропорционально не только площади области  $S$  и плотности  $D$  регулярного покрытия, но и специальному множителю, зависящему от  $P$  и  $\alpha$ .

## 2.2. Построение типовой функции затрат для покрытий произвольной сложности

Пусть регулярное покрытие  $Cov(F : k_1/p_1, k_2/p_2, \dots, k_n/p_n)$  состоит из  $n$  различных кругов. Тогда соотношение между числом кругов соответствующего вида для одного минимального фрагмента определяется пропорцией:

$$(k_1/p_1) : (k_2/p_2) : \dots : (k_n/p_n) = \lambda_1 : \lambda_2 : \dots : \lambda_n.$$

Величина затрат на один набор сенсорных устройств равна соответствующей сумме:  $P = \lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_n P_n$ . Радиусы всех кругов можно определить через радиус «главного» круга:

$$R_1 = R, R_2 = \nu_2 R, \dots, R_n = \nu_n R, (\nu_1 = 1).$$

Пусть в покрытии области  $S$  задействовано  $N$  минимальных фрагментов. Тогда общее число кругов каждого вида определяется так:  $N_1 = \lambda_1 N, N_2 = \lambda_2 N, \dots, N_n = \lambda_n N$ . Далее получаем обобщённый аналог формулы (2) для функции затрат:

$$E = E_0 + NP + NQ = E_0 + N(\lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_n P_n) + N\mu R^\alpha (\lambda_1 + \lambda_2 \nu_2^\alpha + \dots + \lambda_n \nu_n^\alpha).$$

Учитывая, что плотность покрытия равна

$$D = \frac{N\mu R^\alpha (\lambda_1 + \lambda_2 \nu_2^2 + \dots + \lambda_n \nu_n^2)}{S}, N = \frac{SD}{\pi R^2 (\lambda_1 + \lambda_2 \nu_2^2 + \dots + \lambda_n \nu_n^2)}, \quad (4)$$

выразим функцию затрат через радиус основного круга:

$$E = E_0 + \frac{SD(\lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_n P_n)}{\pi R^2 (\lambda_1 + \lambda_2 \nu_2^2 + \dots + \lambda_n \nu_n^2)} + \frac{SD R^{\alpha-2} \mu (\lambda_1 + \lambda_2 \nu_2^\alpha + \dots + \lambda_n \nu_n^\alpha)}{\pi (\lambda_1 + \lambda_2 \nu_2^2 + \dots + \lambda_n \nu_n^2)}. \quad (5)$$

Используя данную функцию затрат, можно доказать следующий результат.

**Теорема 2.** Пусть для мониторинга области с площадью  $S$  используется сенсорная сеть с регулярной структурой покрытия  $Cov(F : k_1/p_1, k_2/p_2, \dots, k_n/p_n)$  плотности  $D$  и определены следующие характеристики:

1. Задано соотношение между радиусами действия сенсорных устройств:

$$R_1 = R, R_2 = \nu_2 R, \dots, R_n = \nu_n R, (\nu_1 = 1);$$

2. Отношение между количествами кругов разного вида определяется пропорцией:

$$\begin{aligned} N_1 : N_2 : \dots : N_n &= (k_1/p_1) : (k_2/p_2) : \dots : (k_n/p_n) = \\ &= \lambda_1 N : \lambda_2 N : \dots : \lambda_n N = \lambda_1 : \lambda_2 : \dots : \lambda_n; \end{aligned}$$

3. Величина затрат на один набор сенсорных устройств равна  $P = \lambda_1 P_1 + \lambda_2 P_2 + \dots + \lambda_n P_n$ , где  $P_1, P_2, \dots, P_n$  - это затраты на соответствующие типы устройств;

4. Затраты на эксплуатацию определяются в соответствии с радиусами действий сенсорных устройств по закону  $Q = \mu(\lambda_1 R_1^\alpha + \lambda_2 R_2^\alpha + \dots + \lambda_n R_n^\alpha)$ ,  $\alpha \in (2, 6)$ .

Тогда оптимальные параметры сенсорной сети определяются следующими соотношениями:

$$\begin{aligned} R &= \left( \frac{2P}{\mu(\alpha-2)(\lambda_1 + \lambda_2 \nu_2^\alpha + \dots + \lambda_n \nu_n^\alpha)} \right)^{1/\alpha}, \\ N &\approx \frac{SD}{\pi} \frac{(\mu(\alpha-2)(\lambda_1 + \lambda_2 \nu_2^\alpha + \dots + \lambda_n \nu_n^\alpha))^{2/\alpha}}{(2P)^{2/\alpha} (\lambda_1 + \lambda_2 \nu_2^2 + \dots + \lambda_n \nu_n^2)}. \quad (6) \end{aligned}$$

*Доказательство.* Исследуя функции (5) на минимум, определяем оптимальное значение переменной  $R$ , приведённое в (6). Используя (4) и проводя ряд преобразований, можно получить требуемое соотношение для  $N$ . ■

**Следствие 2.** *Используя соотношения (6) и формулу (5), можно определить минимальные затраты на проектирование и эксплуатацию сенсорной сети по формуле:*

$$E = E_0 + \frac{SDP(\mu(\alpha - 2)(\lambda_1 + \lambda_2\nu_2^\alpha + \dots + \lambda_n\nu_n^\alpha))^{2/\alpha}}{\pi(\lambda_1 + \lambda_2\nu_2^2 + \dots + \lambda_n\nu_n^2)(2P)^{2/\alpha}} \cdot \left(1 + \frac{2}{(\alpha - 2)(\lambda_1 + \lambda_2\nu_2^\alpha + \dots + \lambda_n\nu_n^\alpha)}\right).$$

### 3. Реализация алгоритма для регулярных структур сложности $L1$ и $L2$

Покажем, как реализуются методы расчёта оптимальных параметров для конкретных регулярных покрытий. Рассмотрим четыре модели покрытий, фрагменты которых изображены на рисунке 4. В покрытиях  $A - 1$  и  $B - 1$  используются круги одного размера (уровень сложности  $L1$ ), а в покрытиях  $A - 2$  и  $B - 2$  используются круги двух различных типов ( $L2$ ).

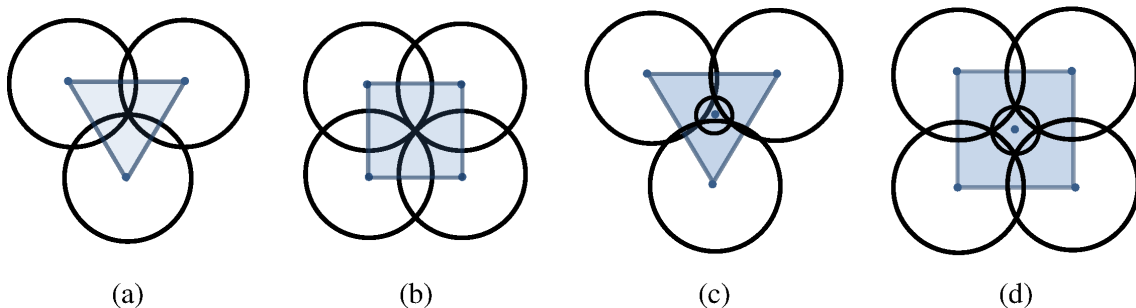


Рис. 4. Элементы покрытий. (a) Модель  $A - 1$ , (b) Модель  $B - 1$ , (c) Модель  $A - 2$ , (d) Модель  $B - 2$

В первой модели центры трёх соседних кругов находятся в вершинах равностороннего треугольника и имеют одну общую точку. Во второй модели центры находятся в вершинах квадрата, и также имеется одна узловая точка для четырёх соседних кругов. Плотность покрытия  $D$  будет определяться отношением суммарной площади  $Sf$  частей окружностей, покрывающих этот фрагмент к площади  $Sp$  типового фрагмента.

Для модели  $A - 1$  получаем

$$Sp = \frac{3\sqrt{3}R^2}{4}, \quad Sf = \frac{\pi R^2}{2}, \quad D_{A-1} = \frac{Sf}{Sp} = \frac{2\pi}{3\sqrt{3}} \approx 1,2092.$$

Для модели  $B - 1$  плотность покрытия существенно выше:

$$Sp = 2R^2, \quad Sf = \pi R^2, \quad D_{B-1} = \frac{Sf}{Sp} = \frac{\pi}{2} \approx 1,5708.$$

Используя значения плотностей  $D_{A-1}$  и  $D_{B-1}$ , можно оптимизировать затраты сенсорных сетей с соответствующими структурами согласно теореме 1. При этом радиус оптимального круга  $R = \left(\frac{2P}{\mu(\alpha-2)}\right)^{1/\alpha}$  не зависит от плотности покрытия, а оптимальное количество кругов определяется соотношениями:

$$N_{A-1} = \frac{SD_{A-1}}{\pi R^2} \approx \frac{2S}{3\sqrt{3}} \left(\frac{\mu(\alpha-2)}{2P}\right)^{2/\alpha}, \quad N_{B-1} = \frac{SD_{B-1}}{\pi R^2} \approx \frac{S}{2} \left(\frac{\mu(\alpha-2)}{2P}\right)^{2/\alpha}.$$

Рассмотрим более подробно модели  $A - 2$  и  $B - 2$ . В работе [3] рассчитаны оптимальные соотношения между радиусами большего и меньшего кругов и соответствующие минимальные плотности.

Оптимальные параметры модели  $A - 2$ :

$$r = \frac{R}{\sqrt{31}} \approx 0,1796R; \quad Sp = \frac{27\sqrt{3}}{31}R^2; \\ Sf = \frac{33}{62}\pi R^2; \quad D_{A-2} = \frac{Sf}{Sp} = \frac{11\pi}{18\sqrt{3}} \approx 1,1084. \quad (7)$$

Оптимальные параметры модели  $B - 2$ :

$$r = \frac{R}{\sqrt{5}} \approx 0,4472R; \quad Sp = \frac{16}{5}R^2; \quad Sf = \frac{6}{5}\pi R^2; \quad D_{B-2} = \frac{Sf}{Sp} = \frac{3\pi}{8} \approx 1,1781. \quad (8)$$

Плотность покрытия модели  $A - 2$  меньше, чем у модели  $B - 2$ , хотя они сравнимы по качеству. Однако модель  $B - 2$  имеет преимущество за счёт более простой структуры решётки центров кругов. Более того, суммарное число кругов менее плотного покрытия  $B - 2$  (больших и маленьких) будет меньше, чем у модели  $A - 2$ . Поэтому при некоторых соотношениях между ценами  $P_1$  и  $P_2$  затраты по модели  $B - 2$  могут быть меньше, чем у модели  $A - 2$ .

Модель  $A - 2$  будет принадлежать классу  $Cov(F : 1_{12}, 1_6)$ . Минимальный фрагмент в этом случае – прямоугольный треугольник с острыми углами по 30 и 60 градусов (рис. 5). Количество больших кругов покрытия в два раза меньше, чем маленьких, так как:  $(1/12) : (1/6) = 1 : 2$ . Для модели покрытия  $B - 2$  класса  $Cov(F : 1_8, 1_8)$  соотношение между количествами разных кругов одинаковое:  $(1/8) : (1/8) = 1 : 1$ .

**Предложение 1.** Пусть для мониторинга области с площадью  $S$  используется сенсорная сеть с регулярной структурой покрытия  $Cov(F : 1_{12}, 1_6)$ , соответствующей модели  $A - 2$ . Тогда оптимальные параметры сенсорной сети определяются следующими соотношениями:  $R_1 = R$ ,  $R_2 = \frac{R}{\sqrt{31}}$ ,  $N_1 = N$ ,

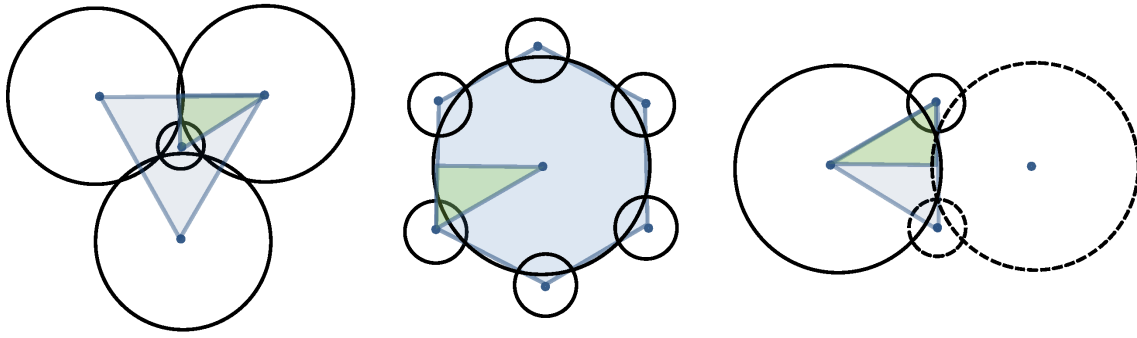


Рис. 5. Варианты плиток и минимальный фрагмент для Модели  $A-2$

$N_2 = 2N$ , где

$$R = \left( \frac{2(P_1 + 2P_2)}{\mu(\alpha - 2)(1 + 2 \cdot 31^{-\alpha/2})} \right)^{1/\alpha},$$

$$N \approx \frac{11S}{18\sqrt{3}} \left( \frac{\mu(\alpha - 2)(1 + 2 \cdot 31^{-\alpha/2})}{2(P_1 + 2P_2)(33/31)} \right)^{2/\alpha}. \quad (9)$$

**Предложение 2.** Пусть для мониторинга области с площадью  $S$  используется сенсорная сеть с регулярной структурой покрытия  $\text{Cov}(F : 1_8, 1_8)$ , соответствующей модели  $B-2$ . Тогда оптимальные параметры сенсорной сети определяются следующими соотношениями:  $R_1 = R$ ,  $R_2 = \frac{R}{\sqrt{5}}$ ,  $N_1 = N_2 = N$ , где

$$R = \left( \frac{2(P_1 + P_2)}{\mu(\alpha - 2)(1 + 5^{-\alpha/2})} \right)^{1/\alpha}, \quad N \approx \frac{3S}{8} \left( \frac{\mu(\alpha - 2)(1 + 5^{-\alpha/2})}{2(P_1 + P_2)(6/5)} \right)^{2/\alpha}. \quad (10)$$

*Доказательство.* В случае модели  $A-2$  можно взять  $\lambda_1 = 1$  и  $\lambda_2 = 2$ . Учитывая соотношение между радиусами (7),  $\nu_2 = 1/\sqrt{31}$ . Далее по теореме 2 при  $n = 2$  получаем требуемое утверждение 1.

В случае модели  $B-2$  можно взять  $\lambda_1 = 1$  и  $\lambda_2 = 1$ . Из соотношений (8) определяем значение  $\nu_2 = 1/\sqrt{5}$ . По теореме 2 получаем доказательство утверждения 2. ■

Приведём пример, когда модель  $B-2$  лучше, чем модель  $A-2$ . При  $E_0 = 0$ ,  $P_1 = P_2 = 1$ ,  $\mu = 1$ ,  $\alpha = 4$ ,  $S = 100$  получаем:

$$R_{A-2} \approx 1,3157, r_{A-2} \approx 0,2363, N_{A-2}^{(1)} = 197, N_{A-2}^{(2)} = 2N_{A-2}^{(1)} = 394, E_{A-2} \approx 1163;$$

$$R_{B-2} \approx 1,1776, r_{B-2} \approx 0,5266, N_{B-2}^{(1)} = N_{B-2}^{(2)} = 247, E_{B-2} \approx 884;$$

#### 4. Численный эксперимент

Проведём вычислительный эксперимент для проверки точности работы алгоритма. В качестве регулярной структуры сенсорной сети выберем модель  $A-2$



с плотностью покрытия из (7):  $D_{A-2} \approx 1,1084$ . Дадим фиксированные значения показателям:  $\alpha = 4, \mu = 1$  и  $P_1 = P_2 = 1$ . Размеры радиусов действия сенсоров, которые не зависят от размеров области мониторинга, вычисляем по формулам (9) и (7):  $R = 1,3157, r = 0,2663$ . Найденные размеры кругов используем для построения регулярной сети на достаточно большом трафарете. Для разных значений площади области будем вычислять теоретические значения числа сенсоров. Выберем пять значений для круговой области  $S$ , соответственно, с диаметрами 12, 14, 16, 18, и 20 (см. таблицу 1). Далее для области с площадью  $S$  (заданной в соответствующем масштабе) выполняем по одной серии из десяти численных экспериментов:

1. Случайным образом накладываем область на трафарет с регулярной структурой (или трафарет на область);
2. Отбираем круги, участвующие в покрытии области, и вычисляем их количество  $N$ ;
3. Полученные данные сравниваем с теоретическими показателями, оцениваем относительное отклонение.

Таблица 1. Данные вычислительного эксперимента

| № \ $S$              | 113,1 | 153,9 | 201,6 | 254,5  | 314,2  |
|----------------------|-------|-------|-------|--------|--------|
| 1                    | 26    | 35    | 42    | 52     | 64     |
| 2                    | 25    | 34    | 43    | 51     | 62     |
| 3                    | 26    | 33    | 44    | 50     | 63     |
| 4                    | 26    | 35    | 40    | 53     | 61     |
| 5                    | 27    | 34    | 41    | 50     | 64     |
| 6                    | 27    | 35    | 42    | 51     | 63     |
| 7                    | 25    | 33    | 42    | 50     | 62     |
| 8                    | 27    | 34    | 43    | 52     | 61     |
| 9                    | 26    | 35    | 45    | 53     | 63     |
| 10                   | 27    | 34    | 43    | 52     | 62     |
| $N_{cp}$             | 26,2  | 34,2  | 42,5  | 51,4   | 62,5   |
| $N_t$                | 23    | 31    | 40    | 51     | 63     |
| $(N_{cp} - N_t)/N_t$ | 0,14  | 0,10  | 0,06  | 0,0078 | 0,0079 |

Для сравнительно небольших областей (относительно радиуса действия сенсоров) ошибка составляет примерно 10%. При увеличении области ошибка быстро уменьшается и становится менее 1%.

Дополнительно можно сравнить вычислительные значения функции затрат при других значениях  $R$  и убедиться в эффективности расчётного алгоритма.

## 5. Заключение

Покрывающая эффективность (coverage efficient) для моделей второго уровня зависит не от размера радиусов кругов, а от их отношения. Поэтому за счёт изменения размера кругов при заданном отношении радиусов можно оптимизировать затраты на эксплуатацию. В данной работе показана возможность реализации предложенного подхода для определения количества сенсорных устройств и выбора оптимальных радиусов действия. Оценка оптимальных размеров кругов зависит от баланса эксплуатационных и монтажных затрат. Если стоимость монтажных работ растёт, то радиус действия сенсорного устройства целесообразно увеличивать. Если же увеличивается стоимость эксплуатационных затрат (коэффициента  $\mu$ ), то радиус уменьшается.

Для проверки адекватности предложенной модели были проведены численные эксперименты. Эксперимент заключался в том, что на заданную область «накладывалось» регулярное покрытие и отбирались круги, участвующие в покрытии области. Сравнительный анализ показал практическое совпадение теоретических и вычислительных значений. Показано, что при увеличении площади области относительная ошибка расчётных показателей уменьшается.

Предложенная методика хорошо работает для выпуклых областей с относительно небольшим периметром. Критерием отклонения может служить величина  $W = P^2/(4\pi S)$ . Для круговой области критерий  $W$  принимает значение 1. Для других же областей эта величина увеличивается в зависимости от «растягивания» области. Заметим, что в любом случае можно получить гарантированную оценку для числа сенсорных устройств, добавив к области слой толщины  $\epsilon$  (порядка сенсорного радиуса). Гарантированная оценка получится, если в расчёты подставить площадь увеличенной области, которая определяется по формуле:  $S^* = S + P\epsilon + \pi\epsilon^2$ . В следующих работах мы собираемся более подробно исследовать эту тему.

## ЛИТЕРАТУРА

1. Wu J., Yang S. Energy-Efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges // Int. J. of Foundations of Computer Science. 2005. N. 6(1). P. 3–17.
2. Cardei M. Improving Network Lifetime using Sensors with Adjustable Sensing Ranges // Int. J. of Sensor Networks. 2006. N. 1. P. 41–49.
3. Астраков С.Н., Ерзин А.И., Залубовский В.В. Сенсорные сети и покрытие плоскости кругами // Дискретный анализ и исследование операций. 2009. № 3. С. 3–19.
4. Zalubovsky V., Astrakov S., Erzin A., Choo H. Energy-efficient Area Coverage by Sensors with Adjustable Ranges // Sensors. 2009. N. 9(4). P. 2446–2460.
5. Zhang H., Hou J.C. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks // Ad Hoc and Sensor Networks. 2005. V. 1. P. 89–124.
6. Hadwiger H. Überdeckungen ebener bereiche durch kreise und quadrate // Comment. Math. Helv. 1940–1941. N. 13. P. 195–200.

7. Erzin A., Astrakov S. Min-Density Stripe Covering and Applications in Sensor Networks // International Conference on Computational Science and Applications, Springer-Verlag, Berlin Heidelberg. 2011. Part III. P. 152–162.
8. Астраков С.Н., Ерзин А.И. Сенсорные сети и покрытие полосы эллипсами // Вычислительные технологии. 2013. Т. 18, № 2. С. 3–17.

## **BUILDING COST-EFFECTIVE SENSOR NETWORKS**

**S.N. Astrakov**<sup>1,2</sup>

Ph.D. (Phys.-Math.), Associate Professor, e-mail: astrakov90@gmail.com

**A.G. Kvashnin**<sup>1</sup>

Ph.D. (Phys.-Math.), e-mail: kvashninag@kti.sbras.ru

**L.A. Korolenko**<sup>1</sup>

Software Engineer, e-mail: lakorolenko@gmail.com

<sup>1</sup>Institute of Computational Technologies, Siberian Branch of Russian Academy of Science

<sup>2</sup>Novosibirsk National Research State University

**Abstract.** In this paper, we present methods for designing wireless sensor networks with the use of models of the regular circular coverage. The problem dealt with in this work is building the most efficient sensor network at a given complexity of its structure. When solving this problem, the cost of a typical sensor and expenses to run the sensor network are taken into account. The size of the effective area of sensors corresponds to radii of circles. Complexity of the structure of a sensor network is defined by the number of different sizes of circles (one, two or more types) and by a method of positioning the circles. Accordingly, principles of classification of the regular coverage are offered which are based on a notion of the minimum fragment. Methods of calculating optimum number of sensors for bounded regions at a given structure of the regular sensor network are presented.

**Keywords:** sensor networks, circular models of coverage, density of coverage, regular structure of coverage, cost optimization.

*Дата поступления в редакцию: 14.03.2017*

## РАВНОВЕСНАЯ ДИНАМИКА ПЛОДОРОДИЯ ПОЧВЫ В ЗАСУШЛИВЫХ И ВЛАЖНЫХ РЕГИОНАХ

Л.А. Володченкова

к.б.н., доцент, e-mail: volodchenkova2007@yandex.ru

А.К. Гуц

д.ф.-м.н., профессор, e-mail: guts@omsu.ru

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** В статье исследуется равновесная в смысле Нэша динамика почвы в засушливых и влажных регионах в рамках теории дифференциальных игр.

**Ключевые слова:** равновесие Нэша, модель почвы, плодородие почвы, засушливый регион, дифференциальные игры.

Почва относится к числу основных факторов, определяющих условия произрастания растительности. В [1–3] была предложена модель почвы в виде дифференциального уравнения для меры плодородия почвы:

$$\frac{dy}{dt} = \gamma \cdot [(p - p_0) - y^2]y - \delta \cdot (W - W_-)(W - W_+), \quad (1)$$

где  $y$  — мера плодородия почвы,  $p$  — мера типа почвообразующей породы,  $W$  — влажность почвы,  $W_-$  — значение влажности почвы, которое характеризует нехватку воды, и, соответственно,  $W_+$  — её избыток,  $0 < W_- < W_+$ , и  $\gamma, \delta$  — положительные константы.

Уравнение системы (1) — это уравнение, реализующее упрощённое представление о плодородии почвы и учитывающее только два фактора: тип почвообразующей породы и влажность почвы. Их изменение может привести к скачкообразному изменению плодородия почвы, и это мы смоделировали, вводя в правую часть уравнения катастрофу типа «сборка». В точке  $(p_0, W_-)$  происходит катастрофа падения плодородия, связанная с нехваткой воды в почве, а в точке  $(p_0, W_+)$  — катастрофа падения плодородия при избытке влаги [1].

Мы видим, что динамика плодородия почвы зависит от двух управляющих факторов: от меры типа почвообразующей породы  $p$  и от влажности почвы  $W$ .

В данной статье мы посмотрим на эти два фактора как на двух игроков, которые пытаются оптимизировать свой выигрыш, меняя стратегии управления.

Очевидно, в природе реализуется некоторое равновесие, которое возникает в том случае, когда один игрок каким-то образом соотносит свои действия, свою стратегию управления с действиями другого игрока.

В теории игр одним из самых известных равновесий является *равновесие Нэша*.

Реализация равновесия Нэша означает, что если каждый игрок пытается в одностороннем порядке изменить свою стратегию управления, в то время как политика остальных игроков остаётся неизменной, то он имеет худший результат (большой проигрыш).

## 1. Динамика плодородия почвы для засушливых регионов

При длительных засухах происходит естественное опустынивание. Опустынивание (дезертификация) почв — это деградация почв с постепенной потерей плодородия.

В засушливых районах влияние растительности на почву ослаблено в силу её скудности. Поэтому имеет смысл изучить динамику почвы, не учитывая взаимодействие почвы с растительностью.

В нашей модели почвы, если речь идёт о засушливых районах, влажность  $W$  находится в окрестности параметра  $W_-$ . Иначе говоря, вместо уравнения (1) будем изучать уравнение

$$\frac{dy}{dt} = -\gamma y^3 + \gamma y p + \mu \cdot (W - W_-), \quad (2)$$

$$\mu > 0, \quad t \in [0, T].$$

## 2. Динамика плодородия почвы для влажных регионов

Однако можно искать равновесия, предполагая, что недостаток влаги (засуха) в наших краях — явление крайне редкое, и поэтому можно считать, что  $W$  находится в окрестности параметра  $W_+$ . Иначе говоря, вместо уравнения (1) можно изучать уравнение

$$\frac{dy}{dt} = -\gamma y^3 + \gamma y p - \lambda \cdot (W - W_+), \quad (3)$$

$$\lambda > 0, \quad t \in [0, T].$$

Почва связана потоками энергии и вещества с растительностью и совместно с животными и микроорганизмами формирует целостный лесной биогеоэкологический покров. Растительность влажных районов — существенная составляющая биоценоза. Поэтому динамику плодородия почвы для влажного региона следует изучать с учётом взаимодействия в системе «почва-растительность». Это было сделано в статье [4].

## 3. Алгоритм нахождения равновесий Нэша

Рассматривать игру с ненулевой суммой вполне разумно, поскольку «выигрыши» наших игроков  $p$  и  $W$  слабо связаны.

Если игрок формирует «свое» управляющее воздействие в виде только функции времени  $u(t)$  на всю продолжительность игры, то  $u(t)$  — это *программное управление* игрока. Ранее мы называли его, используя термин «управление». Однако игрок может выбирать своё управление в зависимости от того, в каком положении  $y$  в момент времени  $t$  находится система «почва». В таком случае игрок конструирует управляющее воздействие в виде функции  $u(t, y)$ , зависящей уже от позиции  $\{t, y\}$ , и для  $u(t, y)$  используется термин *позиционное управление* игрока [5]. Часто пишут просто  $u(y)$ .

Мы будем искать позиционное управление, позиционное равновесие Нэша. Для дифференциальной игры  $N$ -игроков [6]

$$\frac{dz}{dt} = f(z) + \sum_{j=1}^N g_j(z) u_j, \quad z \in \mathbb{R}^n, \quad u_j \in \mathbb{R}^{m_j}, \quad f(0) = 0,$$

$$J_i(z, u_1, \dots, u_N) = \int_0^{+\infty} [Q_i(z) + \sum_{j=1}^N u_j^T R_{ij} u_j] dt, \quad (i = 1, \dots, N),$$

где  $Q_i > 0$  и  $R_{ii} > 0$ ,  $R_{ij} \geq 0$  — симметрические матрицы, *существование равновесий Нэша*

$$J_i(u_1^*, u_2^*, u_i^*, \dots, u_N^*) \leq J_i(u_1^*, u_2^*, \dots, u_{i-1}^*, u_i, u_{i+1}^*, \dots, u_N^*), \quad \forall u_i, \quad i = 1, \dots, N \quad (4)$$

сводится к крайне сложной задаче отыскания решения  $V_i(z)$  нелинейного уравнения Гамильтона-Якоби

$$\begin{aligned} (\nabla V_i)^T f(z) + Q_i(z) - \frac{1}{2} (\nabla V_i)^T \sum_{j=1}^N g_j(z) (R_{jj})^{-1} (g_j(z))^T (\nabla V_j) + \\ + \frac{1}{4} \sum_{j=1}^N (\nabla V_j)^T g_j(z) R_{jj}^{-T} R_{ij} R_{jj}^{-1} (g_j(z))^T (\nabla V_j) = 0, \end{aligned} \quad (5)$$

$$\nabla V_i = \begin{pmatrix} (V_i)'_x \\ (V_i)'_y \end{pmatrix}, \quad (\nabla V_i)^T = ((V_i)'_x, (V_i)'_y),$$

по которому строится равновесие Нэша [6, Theorem 10.4-2, утверждение b.]:

$$u_i^*(z) = u_i(V_i(z)) = -\frac{1}{2} R_{ii}^{-1} (g_i(z))^T (\nabla V_i), \quad i = 1, \dots, N. \quad (6)$$

Равновесная динамика с управлением Нэша (6) задаётся уравнением

$$\frac{dz}{dt} = f(z) - \frac{1}{2} \sum_{j=1}^N g_j(z) R_{jj}^{-1} (g_j(z))^T (\nabla V_j). \quad (7)$$

#### 4. Нэшевское равновесие для уравнений (2) и (3)

Для того чтобы проанализировать динамику плодородия почвы в засушливых и незасушливых районах и вести вычисления одновременно, уравнения (2), (3) запишем в едином виде

$$\frac{dy}{dt} = -\gamma y^3 + \gamma y p - \omega \cdot (W - \widehat{W}), \quad (8)$$

где для засушливого района:  $\omega = -\mu < 0$ ,  $\widehat{W} = W_-$  и  $\omega = \lambda > 0$ ,  $\widehat{W} = W_+$  — для влажного.

$$\lambda > 0, \quad t \in [0, T].$$

Сделаем замену в уравнении (7)  $y = \bar{y} + c$ ,  $c = const$  и подберём  $c$  так, чтобы слагаемое в правой части, в которое не входят факторы  $p, W$  при  $\bar{y} = 0$  обращалось в нуль, как это требуется в теории из § 3.

Легко найти, что  $c = (\omega \widehat{W} / \gamma)^{1/3}$ . В результате такой замены мы вместо уравнения (7) можем изучать, не ограничивая общности, уравнение

$$\frac{dy}{dt} = [-\gamma y^3 - 3\gamma c y^2 - 3\gamma c^2 y - \gamma c^3 + \omega \widehat{W}] + \gamma(y + c)p - \omega W. \quad (9)$$

Уравнение (9) представим в виде

$$\frac{dy}{dt} = f(y) + g_1(y)u_1 + g_2(y)u_2, \quad (10)$$

где

$$f(y) = -\gamma y^3 - 3\gamma c y^2 - 3\gamma c^2 y - \gamma c^3 + \omega \widehat{W}, \quad f(0) = 0,$$

$$g_1(y) = \gamma(y + c), \quad g_2(y) = -\omega,$$

$$u_1 = p, \quad u_2 = W.$$

Будем считать, что у нас 2 игрока. Игрок 1 — это фактор  $u_1 = p$  — мера типа почвообразующей породы, игрок 2 — влажность почвы.

Выигрышные функции возьмём в виде:

$$J_i(y, u_1, u_2) = \int_0^{+\infty} [Q_i(y) + u_i^2] dt, \quad (i = 1, 2) \quad (11)$$

и числа

$$Q_i > 0, \quad R_{ii} > 0, \quad R_{ij} \geq 0.$$

Рассматриваем игру с ненулевой суммой.

Используем теперь теорию, изложенную в § 3, для нахождения равновесий Нэша для уравнения (10).

В нашем случае  $N = 2$ , и рассматриваем  $R_{11} = R_{22} = 1$ ,  $R_{ij} = 0$  ( $i \neq j$ ).

Тогда уравнения Гамильтона-Якоби (5) имеют вид

$$\begin{aligned} Q_1 + (V_1)'f(y) - \frac{1}{2}(V_1)'[g_1^2(V_1)' + g_2^2(V_2)'] - \frac{1}{4}[g_1(y)]^2[(V_1)']^2 &= 0, \\ Q_2 + (V_2)'f(y) - \frac{1}{2}(V_2)'[g_1^2(V_1)' + g_2^2(V_2)'] - \frac{1}{4}[g_2(y)]^2[(V_2)']^2 &= 0. \end{aligned} \quad (12)$$

Полагая, что

$$V_1'(y) = V_2'(y) = -f(y),$$

получаем уравнения Гамильтона-Якоби в виде

$$\begin{aligned} Q_1(y) &= [f(y)]^2(1 + \frac{1}{2}[g_1^2] + \frac{1}{4}[g_2]^2) > 0, \\ Q_2(y) &= [f(y)]^2(1 + \frac{1}{4}[g_1]^2 + \frac{1}{2}[g_2]^2) > 0. \end{aligned} \quad (13)$$

Следовательно, если  $Q_i$  выбрать именно такими, то уравнения Гамильтона-Якоби выполняются.

Поэтому по теореме 10.4-2 (пункт b)) из [6] имеем равновесие Нэша

$$p^*(y) = \frac{1}{2}g_1(y)f(y), \quad W^*(y) = \frac{1}{2}g_2(y)f(y), \quad (14)$$

найденное по формулам (6).

Выигрышные/проигрышные функции поэтому имеют вид:

$$\begin{aligned} J_1(y, p, W) &= \int_0^{+\infty} (Q_1(y) + [p^*(y)]^2) dt, \\ J_2(y, p, W) &= \int_0^{+\infty} (Q_2(y) + [W^*(y)]^2) dt. \end{aligned} \quad (15)$$

Равновесная нэшевская динамика описывается уравнением (7), имеющим в нашем случае следующий вид

$$\frac{dy}{dt} = f(y) \left( 1 + \frac{1}{2}[g_1(y)]^2 + \frac{1}{2}[g_2(y)]^2 \right), \quad (16)$$

или

$$\frac{dy}{dt} = \left( -\gamma y^3 - 3\gamma c y^2 - 3\gamma c^2 y - \gamma c^3 + \omega \widehat{W} \right) \cdot \left( 1 + \frac{1}{2}[\gamma(y+c)]^2 + \frac{1}{2}\omega^2 \right), \quad (17)$$

получаемым из уравнения (9) заменой  $p$  и  $W$  на  $p^*(y)$  и  $W^*(y)$  соответственно.

Рассмотрим случай засушливого региона и возьмём

$$\gamma = 1, \quad \omega = -1, \quad \widehat{W} = 1, \quad c = -1. \quad (18)$$



Тогда интегрируя уравнение (17), получаем

$$\int \frac{dy}{\left(-\gamma y^3 - 3\gamma c y^2 - 3\gamma c^2 y - \gamma c^3 + \omega \widehat{W}\right) \cdot \left(1 + \frac{1}{2}[\gamma(y+c)]^2 + \frac{1}{2}\omega^2\right)} = t + const,$$

или

$$\begin{aligned} t = & -1/6 * \ln(y) + 4/21 * \ln(y^2 - 3 * y + 3) - \\ & -4/21 * \sqrt{3} * \arctan(1/3 * (2 * y - 3) * \sqrt{3}) - 3/28 * \ln(4 + y^2 - 2 * y) - \\ & -1/42 * \sqrt{3} * \arctan(1/6 * (2 * y - 2) * \sqrt{3}) + const. \end{aligned}$$

Результат интегрирования уравнения представлен на рис. 1 Мы видим, что плодородие, как и следовало ожидать для засушливого района, падает. Происходит опустынивание почвы.

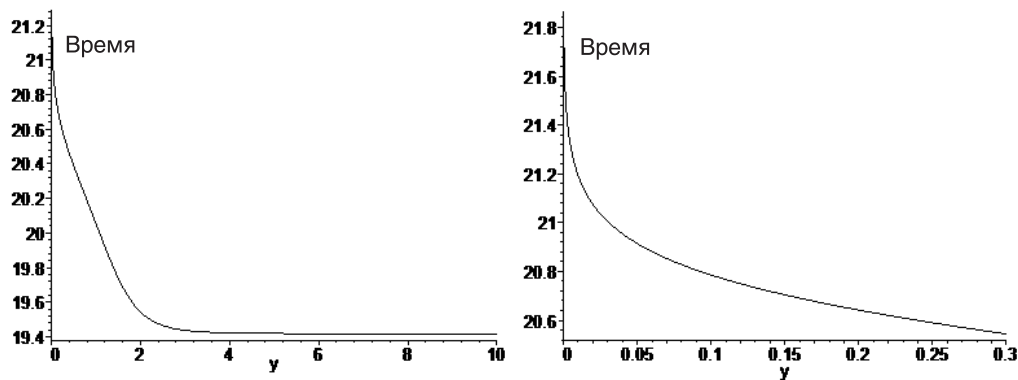


Рис. 1. Примеры равновесной динамики меры плодородия для засушливого региона в условиях равновесия Нэша при  $const = 20$

Обратим внимание, что в нашем примере (18)

$$f(y) = -y^3 + 3y^2 - 3y < 0 \quad \text{при } y > 0,$$

и, следовательно,

$$V_1(y) = V_2(y) = - \int_0^y f(y) dy > 0 \quad \text{при } y > 0.$$

Тогда на основании теоремы 10.4–2 (пункт а)) из [6] можно заявить, что решения уравнения (17), (18) будут асимптотически устойчивыми.

## ЛИТЕРАТУРА

1. Володченкова Л.А. Модель плодородия почвы с точки зрения катастрофы «сборка» // Математическое и компьютерное моделирование: сборник материалов Международной научной конференции (Омск, 21 ноября 2014 г.). Омск : изд-во Ом. гос. ун-та, 2014. С. 25–26.

2. Гуц А.К., Володченкова Л.А. Математическая модель взаимосвязи «растительность-почва» в лесных экосистемах // Математические структуры и моделирование. 2015. № 3(35). С. 56–60.
3. Гуц А.К., Володченкова Л.А. Теоретико-катастрофическая модель взаимосвязи «растительность-почва» в лесных экосистемах // Математическое и компьютерное моделирование: сборник материалов Международной научной конференции (Омск, 21 ноября 2014 г.). Омск : изд-во Ом. гос. ун-та, 2014. С. 23–24.
4. Володченкова Л.А., Гуц А.К. Равновесная динамика лесных экосистем с учетом взаимосвязи «растительность-почва» // Математические структуры и моделирование. 2017. № 2(42). С. 68–79.
5. Тынянский Н.Т., Жуковский В.И. Дифференциальные игры с ненулевой суммой (кооперативный вариант) // Итоги науки и техн. Сер. Мат. анализ. 1979. Т. 17. С. 3–112.
6. Lewis F.L., Vrabie D.L., Syrmos V.L. Optimal control. New Jersey : John Wiley & Sons, Inc., 2012. 540 p.

## THE EQUILIBRIUM DYNAMICS OF SOIL FERTILITY IN ARID AND HUMID REGIONS

**L.A. Volodchenkova**

Ph.D. (Biology), Associate Professor, e-mail: volodchenkova2007@yandex.ru

**A.K. Guts**

Dr.Sc. (Phys.-Math.), Professor, e-mail: guts@omsu.ru

Dostoevsky Omsk State University

**Abstract.** The article investigates the equilibrium dynamics in the sense of Nash of soils in arid and humid regions in the framework of the differential games theory.

**Keywords:** the Nash equilibrium, model of soil, soil fertility, arid region, differential games.

*Дата поступления в редакцию: 11.06.2017*

## **ПРИМЕНЕНИЕ ИНТЕГРАЛЬНОЙ ФОРМЫ УРАВНЕНИЙ КОЛЛОКАЦИЙ И ДИФФЕРЕНЦИАЛЬНЫХ УСЛОВИЙ СОГЛАСОВАНИЯ В МЕТОДЕ КОЛЛОКАЦИЙ И НАИМЕНЬШИХ КВАДРАТОВ**

**Е.В. Ворожцов<sup>1</sup>**

профессор, д.ф.-м.н., e-mail: vorozh@itam.nsc.ru

**В.П. Шапеев<sup>1,2</sup>**

профессор, д.ф.-м.н., e-mail: vshapeev@ngs.ru

<sup>1</sup>Институт теоретической и прикладной механики СО РАН им. С.А. Христиановича

<sup>2</sup>Новосибирский национальный исследовательский университет

**Аннотация.** С целью повышения точности расчётов по методу коллокаций и наименьших квадратов (КНК) предлагается увеличить число степеней свободы с помощью следующих двух способов: увеличения числа базисных функций и интегрирования линеаризованных уравнений с частными производными (УЧП) по подъячейкам каждой ячейки пространственной расчётной сетки. Показано, что предложенные новые варианты метода КНК обладают более высокой точностью, чем предыдущие версии этого метода. Кроме того, вариант метода КНК, который использует интегральную форму уравнений коллокаций, требует для сходимости меньшего числа итераций, чем «дифференциальный» метод КНК.

**Ключевые слова:** метод коллокаций и наименьших квадратов, преобуславливание, подпространства Крылова, многосеточные алгоритмы, коллокация интегральных соотношений, уравнения Навье-Стокса.

### **Введение**

В настоящее время широко применяется численное моделирование различных процессов в технике и промышленности с помощью численного решения начально-краевых задач для нелинейных уравнений с частными производными (УЧП). В частности, некоторые прикладные задачи, в которых необходимо решать уравнения Навье-Стокса, являются настолько вычислительно трудоёмкими, что требуют для своего решения на суперкомпьютерах от нескольких недель до одного года [1, 2]. В этой связи является актуальной, как и прежде, разработка более эффективных методов численного решения систем УЧП, которые позволили бы заметно снижать требуемое машинное время.

Метод КНК, предложенный в [3], получил дальнейшее развитие в последующих работах других авторов, и в настоящее время он является одним

из методов, позволяющих эффективно решать системы уравнений с частными производными, возникающие в разнообразных приложениях [4–13]. Были построены варианты метода, которые позволили получать численные решения двумерных и трёхмерных эталонных задач гидродинамики, по своей точности не уступающие наиболее точным решениям, известным в настоящее время и полученным другими методами [14, 15].

В течение последних трёх десятилетий за рубежом получает всё большее распространение класс численных методов LSFEM (Least-Squares Finite Element Method) — «метод конечных элементов и наименьших квадратов» [16–19]. В нём метод конечных элементов (МКЭ) комбинируется с методом наименьших квадратов. В LSFEM подлежащие решению уравнения с частными производными сначала интегрируются в МКЭ по каждому конечному элементу, который представляет собой подобласть пространственной расчётной области. По аналогии с LSFEM здесь рассмотрим следующие варианты метода КНК: (i) вариант, в котором все уравнения коллокаций, получаемые из системы УЧП, заменяются на их интегральные аналоги, получаемые при интегрировании УЧП по нескольким подъячейкам, на которые разбивается каждая ячейка расчётной области; (ii) вариант, в котором используются и уравнения коллокаций, получаемые из решаемых УЧП, и уравнения, получаемые путём интегрирования УЧП по подъячейкам. Рассматриваемые варианты метода КНК предложены впервые.

Ниже описываются и исходный «дифференциальный» вариант метода КНК, и методы КНК, в которых «дифференциальный» вариант комбинируется с интегральной формой уравнений коллокаций и дифференциальной формой условий согласования. Представлены примеры расчётов, которые показывают, что новые модификации метода КНК позволяют получать более точные результаты, чем в случае «дифференциальных» вариантов метода КНК.

## 1. «Дифференциальный» метод КНК

### 1.1. Описание метода

Рассмотрим краевую задачу для системы уравнений Навье-Стокса

$$(\mathbf{V} \cdot \nabla) \mathbf{V} + \nabla p = \frac{1}{\text{Re}} \Delta \mathbf{V} - \mathbf{f}, \quad \text{div } \mathbf{V} = 0, \quad (x_1, x_2) \in \Omega, \quad (1)$$

$$\mathbf{V}|_{\partial\Omega} = \mathbf{g} \quad (2)$$

в области  $\Omega$  с границей  $\partial\Omega$ . В уравнениях (1)  $x_1, x_2$  — декартовы пространственные координаты;  $\mathbf{V} = (v_1(x_1, x_2), v_2(x_1, x_2))$  — вектор скорости;  $p = p(x_1, x_2)$  — давление;  $\mathbf{f} = (f_1, f_2)$  — заданная вектор-функция;  $\text{Re}$  — число Рейнольдса,  $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$ ,  $(\mathbf{V} \cdot \nabla) = v_1 \frac{\partial}{\partial x_1} + v_2 \frac{\partial}{\partial x_2}$ . Система (1) решается при граничных условиях Дирихле (2), где  $\mathbf{g} = \mathbf{g}(x_1, x_2) = (g_1, g_2)$  — заданная вектор-функция. Давление определяется из (1), (2) с точностью до постоянной. В дальнейшем будем подбирать эту постоянную так, чтобы выполнялось условие

$$\iint_{\Omega} p \, dx_1 dx_2 = 0. \quad (3)$$

В качестве области решения берётся квадрат

$$\Omega = \{(x_1, x_2), 0 \leq x_i \leq L, i = 1, 2\}, \quad (4)$$

где  $L > 0$  — заданная длина стороны квадрата. Величина  $L$  использовалась в конкретных расчётах в качестве характерной длины при обезразмеривании переменных, и она входит естественным образом в определение числа Рейнольдса  $Re$  в (1). Далее краевую задачу для УЧП будем называть дифференциальной задачей.

В данной задаче (1)–(4) область (4) покрывается сеткой из квадратных ячеек  $\Omega_{ij}$ ,  $i, j = 1, \dots, I$ ,  $I \geq 1$ . Удобно ввести локальные координаты  $y_1, y_2$  в каждой ячейке  $\Omega_{ij}$ . Зависимость локальных координат от глобальных координат  $x_1, x_2$  задаётся формулами  $y_m = (x_m - x_{m,i,j})/h$ ,  $m = 1, 2$ , где  $x_{m,i,j}$  — значение координаты  $x_m$  в центре ячейки  $\Omega_{ij}$ , а  $h$  — половина длины стороны квадратной ячейки,  $h = L/(2M)$ . Пусть  $\mathbf{u}(y_1, y_2) = (u_1, u_2) = \mathbf{V}(hy_1 + x_{1,i,j}, hy_2 + x_{2,i,j})$ ,  $q(y_1, y_2) = p(hy_1 + x_{1,i,j}, hy_2 + x_{2,i,j})$ . В локальных переменных уравнения Навье-Стокса принимают следующий вид:

$$\Delta u_m - Re \, h \left( u_1 \frac{\partial u_m}{\partial y_1} + u_2 \frac{\partial u_m}{\partial y_2} + \frac{\partial q}{\partial y_m} \right) = Re \cdot h^2 f_m, \quad m = 1, 2; \quad (5)$$

$$\frac{1}{h} \left( \frac{\partial u_1}{\partial y_1} + \frac{\partial u_2}{\partial y_2} \right) = 0, \quad (6)$$

где  $\Delta = \frac{\partial^2}{\partial y_1^2} + \frac{\partial^2}{\partial y_2^2}$ . Линеаризация по Ньютону уравнений (5) приводит к формуле

$$\xi [\Delta u_m^{s+1} - (Re \cdot h)(u_1^s u_{m,y_1}^{s+1} + u_1^{s+1} u_{m,y_1}^s + u_2^s u_{m,y_2}^{s+1} + u_2^{s+1} u_{m,y_2}^s + q_{y_m}^{s+1})] = \xi F_m, \quad (7)$$

где  $m = 1, 2$ , а  $s$  — номер итерации по нелинейности,  $s = 0, 1, 2, \dots$ ,  $u_1^s, u_2^s, q^s$  — известное приближение решения на  $s$ -ой итерации начиная с выбранного начального приближения с индексом  $s = 0$ ,  $F_m = Re [h^2 f_m - h(u_1^s u_{m,y_1}^s + u_2^s u_{m,y_2}^s)]$ ,  $u_{m,y_l} = \partial u_m / \partial y_l$ ,  $q_{y_m} = \partial q / \partial y_m$ ,  $l, m = 1, 2$ . Здесь, как и в [12, 13], введён задаваемый пользователем параметр  $\xi$  с целью управления величиной числа обусловленности переопределённой системы линейных алгебраических уравнений (СЛАУ), которая должна решаться в каждой ячейке  $\Omega_{ij}$ .

Приближённое решение в каждой ячейке  $\Omega_{ij}$  ищется в виде линейной комбинации базисных вектор-функций  $\varphi_l$ :

$$(u_1^s, u_2^s, q^s)^T = \sum_{l=1}^{m_b} b_{i,j,l}^s \varphi_l, \quad (8)$$

где верхний индекс  $T$  обозначает операцию транспонирования, а  $m_b$  — задаваемое пользователем количество базисных вектор-функций. В рассматриваемом варианте метода  $\varphi_l$  являются многочленами. Таким образом, искомое приближенное глобальное решение задачи является кусочно-полиномиальным. А кусок решения (8) в каждой ячейке представляет собой локальное решение в окрестности начала локальной системы координат в ячейке.

Таблица 1. Вид базисных функций  $\varphi_l$ 

| $l$         | 1 | 2      | 3     | 4          | 5          | 6       | 7 | 8     | 9       | 10 | 11    | 12    | 13      | 14       | 15      |
|-------------|---|--------|-------|------------|------------|---------|---|-------|---------|----|-------|-------|---------|----------|---------|
| $\varphi_l$ | 1 | $y_1$  | $y_2$ | $y_1^2$    | $-2y_1y_2$ | $y_2^2$ | 0 | 0     | 0       | 0  | 0     | 0     | 0       | 0        | 0       |
|             | 0 | $-y_2$ | 0     | $-2y_1y_2$ | $y_2^2$    | 0       | 1 | $y_1$ | $y_1^2$ | 0  | 0     | 0     | 0       | 0        | 0       |
|             | 0 | 0      | 0     | 0          | 0          | 0       | 0 | 0     | 0       | 1  | $y_1$ | $y_2$ | $y_1^2$ | $y_1y_2$ | $y_2^2$ |

В работе [12] для аппроксимации составляющих скорости использованы многочлены второй степени по переменным  $y_1, y_2$ , а для аппроксимации давления — многочлены первой степени, так что общее количество базисных вектор-функций в (8) составляло 12.

Ранее в [6] было показано, что можно повысить точность численного решения, получаемого по методу КНК, путём использования многочленов более высоких степеней. В связи с этим мы ниже применяем многочлен второй степени также для аппроксимации давления. В этом случае имеются в общей сложности 18 базисных функций в выбранном пространстве. Поскольку в уравнении неразрывности коэффициенты константы, то его легко удовлетворить за счёт выбора базисных полиномов  $\varphi_l$ . Нетрудно установить, что для этого требуется удовлетворить ими три линейных соотношения. В итоге из первоначальных 18 базисных полиномов независимыми останутся только 15. Они приведены в табл. 1. Их совокупность можно назвать соленоидальным базисом, так как  $\operatorname{div} \varphi_l = 0$ . Таким образом, достоинством предложенного варианта метода является то, что уравнение неразрывности и, следовательно, закон сохранения массы удовлетворяется численным решением задачи во всей расчётной области. Отметим, что проблема обеспечения закона сохранения массы представляет собой существенную трудность для метода LSFEM [19].

Набор базисных функций, применявшийся в [12], получается из набора, представленного в табл. 1, если положить  $m_b = 12$  в (8), то есть если оставить в табл. 1 только первые 12 базисных вектор-функций.

Выпишем систему уравнений, определяющую приближённое решение в каждой ячейке. Назовём её далее локальной СЛАУ, а объединение всех локальных СЛАУ — глобальной СЛАУ. Основными уравнениями, определяющими решение приближённой задачи, являются уравнения коллокаций. Они получаются в результате подстановки в уравнения (7) выражений (8) и координат точек коллокаций. Количество этих точек и их расположение внутри ячейки может варьироваться в различных вариантах метода. В данной работе были реализованы три варианта задания координат точек коллокаций. Обозначим через  $N_c$  число точек коллокации внутри каждой ячейки. При  $N_c = 2$  локальные координаты точек коллокаций таковы:  $(\omega, \omega)$ ,  $(-\omega, \omega)$ , где  $\omega$  — задаваемое пользователем значение в интервале  $0 < \omega < 1$ . При  $N_c = 4$  локальные координаты точек коллокаций имеют вид  $(\pm\omega, \pm\omega)$ . В случае  $N_c = 8$  координаты точек коллокаций задавались следующим образом: расположение первых четырёх точек было взято таким же, как при  $N_c = 4$ , а координаты следующих

четырёх точек задавались по формулам  $(\pm\omega, 0)$ ,  $(0, \pm\omega)$ .

Подставляя (8), а также численные значения координат каждой точки коллокации в (7), получим  $2N_c$  линейных алгебраических уравнений:

$$\sum_{m=1}^{m_b} a_{\nu,m}^{(1)} \cdot b_m^{s+1} = f_{\nu}^s, \quad \nu = 1, \dots, 2N_c. \quad (9)$$

Следуя [12], дополним систему уравнений приближённой задачи в ячейке  $\Omega_{ij}$  линейными условиями согласования локального решения в каждой ячейке с локальными решениями, взятыми во всех соседних с ней ячейках. Эти условия записываются в отдельных точках (называемых точками согласования) на сторонах ячейки  $\Omega_{ij}$ , которые являются общими с соседними ячейками. Условия согласования берутся здесь в виде

$$h \frac{\partial(u^+)^n}{\partial n} + \eta(u^+)^n = h \frac{\partial(u^-)^n}{\partial n} + \eta(u^-)^n, \quad (10)$$

$$h \frac{\partial(u^+)^{\tau}}{\partial n} + (u^+)^{\tau} = h \frac{\partial(u^-)^{\tau}}{\partial n} + (u^-)^{\tau}, \quad (11)$$

$$q^+ = q^-. \quad (12)$$

Здесь  $h \frac{\partial}{\partial n} = h \left( n_1 \frac{\partial}{\partial x_1} + n_2 \frac{\partial}{\partial x_2} \right) = n_1 \frac{\partial}{\partial y_1} + n_2 \frac{\partial}{\partial y_2}$ ,  $n = (n_1, n_2)$  — внешняя нормаль к стороне ячейки  $\Omega_{ij}$ ,  $(\cdot)^n$ ,  $(\cdot)^{\tau}$  — нормальная и касательная составляющие вектора скорости по отношению к стороне ячейки,  $u^+$ ,  $u^-$  — пределы функции  $u$  и  $q$  при стремлении аргументов к точке согласования изнутри и снаружи ячейки  $\Omega_{i,j}$ . Здесь, как и в [12], введён задаваемый пользователем параметр  $\eta$  с целью управления величиной числа обусловленности матрицы СЛАУ, которая должна решаться в каждой ячейке  $\Omega_{ij}$ .

Для однозначного определения давления в решении задаём его значение в одной точке области либо аппроксимируем условие (3) по формуле

$$\frac{1}{h} \left( \iint_{\Omega_{i,j}} q \, dy_1 dy_2 \right) = \frac{1}{h} \left( -I^* + \iint_{\Omega_{i,j}} q^* \, dy_1 dy_2 \right). \quad (13)$$

Здесь  $I^*$  — интеграл по всей области  $\Omega$ , рассчитанный как сумма интегралов по каждой ячейке на предыдущей итерации,  $q^*$  — давление в ячейке на предыдущей итерации.

Обозначим через  $N_m$  число точек согласования для составляющих вектора скорости на сторонах каждой ячейки. При  $N_m = 4$  координаты этих точек согласования задаются формулами  $(\pm 1, 0)$ ,  $(0, \pm 1)$ . При  $N_m = 8$  координаты точек согласования таковы:  $(\pm 1, -\zeta)$ ,  $(\pm 1, \zeta)$ ,  $(-\zeta, \pm 1)$ ,  $(\zeta, \pm 1)$ , где  $0 < \zeta < 1$ . В расчётах, результаты которых представлены ниже, использовалось значение  $\zeta = 1/2$ . Условия согласования для давления (12) задаются в четырёх точках с координатами  $(\pm 1, 0)$ ,  $(0, \pm 1)$ . Используя (8), подставим координаты этих точек в каждое из трёх условий согласования (10)–(12). Из первых двух условий получим  $2N_m$  линейных алгебраических уравнений для составляющих скорости. Подстановка представления (8) в (12) даёт ещё четыре линейных алгебраических уравнения согласования.

В настоящей работе давление задавалось в нижней левой вершине ячейки  $\Omega_{1,1}$  или же использовалось условие (13). Если сторона ячейки лежит на границе области  $\Omega$ , то в соответствующих точках на стороне ячейки вместо условий

согласования в локальной СЛАУ выписываются граничные условия:  $u_m = g_m$ ,  $m = 1, 2$ .

Объединяя уравнения коллокаций, согласования и уравнения, полученные из краевых условий, если ячейка  $\Omega_{i,j}$  граничная, в каждой ячейке получим СЛАУ вида

$$A_{i,j} \cdot \vec{X}_{i,j}^{s+1} = \vec{f}_{i,j}^{s,s+1}, \quad (14)$$

где  $\vec{X}_{i,j}^{s+1} = (b_{i,j,1}^{s+1}, \dots, b_{i,j,m_b}^{s+1})^T$ . В вариантах метода КНК, описываемых в настоящей работе, система (14) является переопределённой.

Из всех возможных псевдорешений полученной переопределённой СЛАУ приближённой задачи здесь наиболее интересное, на котором достигается минимум евклидовой нормы невязки её уравнений. Возникшую задачу наименьших квадратов линейной алгебры (терминология, использованная в [20]) решим сочетанием способа  $QR$ -разложения матриц локальных СЛАУ и итерационным методом решения СЛАУ приближённой задачи. Применение для этого способа  $QR$ -разложения матриц локальных СЛАУ в случае плохо обусловленных задач предпочтительнее, чем способа нормальных уравнений [9, 13, 20]. Для решения СЛАУ приближённой задачи применялся процесс, который условно можно назвать итерациями Гаусса-Зейделя. Одна глобальная  $(s+1)$ -ая итерация заключалась в том, что в расчётной области  $\Omega$  последовательно перебирались все ячейки. При этом в каждой ячейке локальное СЛАУ (14) решалось ортогональным методом (Гивенса или Хаусхолдера), в правой части уравнений (10), (12) в качестве  $u^-$ ,  $q^-$  берутся либо значения решения на  $(s+1)$ -ой итерации, если они уже сосчитаны на этой итерации, либо их значения на предыдущей итерации.

## 1.2. Предобуславливатели для метода КНК

В каждой ячейке  $\Omega_{i,j}$  необходимо решать СЛАУ вида (14). Опустим в (14), для краткости, верхние и нижние индексы:

$$A\vec{X} = \vec{f}. \quad (15)$$

Число обусловленности прямоугольной матрицы  $A$  вычисляется по формуле

$$\kappa(A) = \sqrt{\|A_1\| \cdot \|A_1^{-1}\|}, \quad (16)$$

где предполагается, что квадратная матрица  $A_1 = A^T A$  несингулярная. В нашем случае предобуславливатель зависит от параметров  $\xi, \eta$ . В [12] был описан простой алгоритм для нахождения оптимальных значений  $\xi_{\text{opt}}, \eta_{\text{opt}}$  в любой ячейке из требования минимизации числа обусловленности  $\kappa(\xi, \eta)$ . Число  $\kappa(\xi_{\text{opt}}, \eta_{\text{opt}})$  обычно удовлетворяло неравенствам  $3 < \kappa(\xi_{\text{opt}}, \eta_{\text{opt}}) < 10$ . Оказалось, что оптимальные значения  $\xi_{\text{opt}}, \eta_{\text{opt}}$  слабо зависят от положения конкретной ячейки в расчётной сетке, по меньшей мере, в случаях тех тестовых и эталонных задач, которые рассматривались в [12]. Некоторые свойства предобуславливателя были также исследованы в [12]. В частности, было показано, что снижение  $N_c$  сильнее влияет на значение  $\xi_{\text{opt}}$ , чем на значение  $\eta_{\text{opt}}$ .



### 1.3. Алгоритм ускорения сходимости на основе подпространств Крылова

Для ускорения сходимости итераций здесь применялся во всех новых вариантах метода КНК, описываемых в настоящей работе, новый вариант известного метода [21], основанный на подпространствах Крылова и подробно изложенный ранее в [9, 13]. Ниже кратко описывается соответствующий алгоритм. Пусть СЛАУ имеет вид  $\vec{X} = T\vec{X} + \vec{f}$ , где вектор  $\vec{X}$  — искомое решение,  $T$  — квадратная матрица, а  $\vec{f}$  — вектор-столбец. Пусть матрица  $T$  имеет полный ранг, и пусть сходится следующий итерационный процесс:  $\vec{X}^{n+1} = T\vec{X}^n + \vec{f}$ ,  $n = 0, 1, \dots$ , где  $\vec{X}^n$  — приближение решения на  $n$ -ой итерации. По определению,  $\vec{r}^n = T\vec{X}^n + \vec{f} - \vec{X}^n = \vec{X}^{n+1} - \vec{X}^n$  — невязка уравнений  $\vec{X} = T\vec{X} + \vec{f}$ , и нетрудно получить следующее соотношение из вышеприведённых формул:  $\vec{r}^{n+1} = T\vec{r}^n$ . Предположим, что сделано  $k + 1$  итераций, начиная с начального приближения  $\vec{X}^0$ , то есть вычислены величины  $\vec{X}^1, \vec{X}^2, \dots, \vec{X}^{k+1}$  и  $\vec{r}^0, \vec{r}^1, \dots, \vec{r}^k$ . Значение  $\vec{X}^{k+1}$  затем уточняется по формуле  $\vec{X}^{*k+1} = \vec{X}^{k+1} + \vec{Y}^{k+1}$ . Используется поправка вида

$$\vec{Y}^{k+1} = \sum_{i=1}^k \alpha_i \vec{r}^i \quad (17)$$

с неопределёнными коэффициентами  $\alpha_1, \dots, \alpha_k$ , которые находятся из условия минимизации функционала невязки  $\Phi(\alpha_1, \dots, \alpha_k) = \|\vec{X}^{*k+1} - T\vec{X}^{*k+1} - \vec{f}\|_2^2$ , возникающей при подстановке  $\vec{X}^{*k+1}$  в систему  $\vec{X} = T\vec{X} + \vec{f}$ . Здесь  $\|\vec{u}\|_2$  — евклидова норма вектора  $\vec{u}$  размерности  $N$ . Уточнённый вектор  $k+1$ -го приближения  $\vec{X}^{*k+1}$  используется в качестве начального приближения для дальнейшего продолжения последовательности итераций.

### 1.4. Ускорение сходимости итераций с помощью многосеточного алгоритма

Основная идея многосеточных алгоритмов состоит в селективном демпфировании гармоник ошибки [22, 23]. В методе КНК, как и в других методах, количество итераций, необходимых для достижения заданной точности приближения к предельному решению, зависит от начального приближения. В работе применялась операция продолжения вдоль восходящей ветви V-цикла — расчёты на последовательности измельчающихся сеток — в качестве способа получения хорошего начального приближения для итераций на самой мелкой сетке среди сеток, используемых в многосеточном комплексе. Проиллюстрируем алгоритм операции продолжения на примере составляющей скорости  $u_1(y_1, y_2, b_1, \dots, b_{15})$ . Пусть  $h_1 = h$ , где  $h$  — полушаг грубой сетки, и пусть  $h_2 = h_1/2$  — полушаг мелкой сетки, на которой нужно найти разложение функции  $u_1$  по базису.

*Шаг 1.* Пусть  $X_1$  и  $X_2$  — глобальные координаты центра ячейки грубой сетки. Сделаем следующие подстановки в полиномиальное выражение для  $u_1$ :

$$y_l = (x_l - X_l)/h_l, \quad l = 1, 2. \quad (18)$$

В результате получаем многочлен

$$U_1(x_1, x_2, b_1, \dots, b_{15}) = u_1 \left( \frac{x_1 - X_1}{h_1}, \frac{x_2 - X_2}{h_1}, b_1, \dots, b_{15} \right). \quad (19)$$

*Шаг 2.* Пусть  $(\tilde{X}_1, \tilde{X}_2)$  — глобальные координаты центра любой из четырёх ячеек мелкой сетки, содержащихся в ячейке грубой сетки. Сделаем в (19) замену  $x_l = \tilde{X}_l + \tilde{y}_l \cdot h_2$ ,  $l = 1, 2$ . В результате получим многочлен второй степени  $\tilde{U}_1 = P(\tilde{y}_1, \tilde{y}_2, \tilde{b}_1, \dots, \tilde{b}_{15})$  от переменных  $\tilde{y}_1, \tilde{y}_2$  с коэффициентами  $\tilde{b}_1, \dots, \tilde{b}_{15}$ . После приведения подобных оказывается, что координаты  $X_1, X_2$  и  $\tilde{X}_1, \tilde{X}_2$  входят в  $\tilde{b}_l$  ( $l = 1, \dots, 15$ ) только в виде комбинаций  $\delta x_l = (X_l - \tilde{X}_l)/h_1$ . Согласно (18), величина  $-\delta x_l = (\tilde{X}_l - X_l)/h_1$  является локальной координатой в ячейке грубой сетки центра ячейки мелкой сетки.

Приведём выражения для коэффициентов  $\tilde{b}_j$  ( $j = 1, \dots, 15$ ) представления решения в ячейке мелкой сетки с полушагом  $h_2$  в терминах коэффициентов  $b_1, \dots, b_{15}$  представления решения в ячейке с полушагом  $h_1 = 2h_2$ :

$$\begin{aligned} \tilde{b}_1 &= b_1 - \delta x_1(b_2 - b_4\delta x_1) - \delta x_2(b_3 + 2b_5\delta x_1 - b_6\delta x_2), \quad \tilde{b}_2 = \sigma_1(T_1 + b_5\delta x_2), \\ \tilde{b}_3 &= \sigma_1[b_3 + 2(b_5\delta x_1 - b_6\delta x_2)], \quad \tilde{b}_4 = \sigma_2b_4, \quad \tilde{b}_5 = \sigma_2b_5, \quad \tilde{b}_6 = \sigma_2b_6, \\ \tilde{b}_7 &= b_7 - \delta x_1(b_8 - b_9\delta x_1) + \delta x_2T_1, \quad \tilde{b}_8 = \sigma_1(b_8 - 2b_9\delta x_1 + 2b_4\delta x_2), \\ \tilde{b}_9 &= \sigma_2b_9, \quad \tilde{b}_{10} = b_{10} - \delta x_1T_2 - \delta x_2(b_{12} - b_{15}\delta x_2), \quad \tilde{b}_{11} = \sigma_1(T_2 - b_{13}\delta x_1), \\ \tilde{b}_{12} &= \sigma_1(b_{12} - b_{14}\delta x_1 - 2b_{15}\delta x_2), \quad \tilde{b}_{13} = \sigma_2b_{13}, \quad \tilde{b}_{14} = \sigma_2b_{14}, \quad \tilde{b}_{15} = \sigma_2b_{15}, \end{aligned}$$

где  $\sigma_1 = h_2/h_1$ ,  $\sigma_2 = \sigma_1^2$ ,  $T_1 = b_2 - 2b_4\delta x_1 + b_5\delta x_2$ ,  $T_2 = b_{11} - b_{13}\delta x_1 - b_{14}\delta x_2$ . Заметим, что приведённые выше выражения для  $\tilde{b}_1, \dots, \tilde{b}_9$  совпадают с выражениями, представленными в [12, 13] для случая, когда  $m_b = 12$  в (8).

## 2. Использование интегральной формы уравнений коллокаций

Выше в разделе 1 был описан «дифференциальный» вариант метода КНК, в котором уравнения коллокаций (7) были получены из дифференциальных уравнений (5). Здесь можно использовать вместо уравнений коллокаций (7) их интегральные аналоги, получаемые путём интегрирования уравнений (5) по нескольким подобластям. В «дифференциальном» варианте метода КНК областью влияния локального решения являются точки коллокации и точки согласования. В интегральном варианте областью влияния является множество всех точек ячейки. В этом варианте можно ожидать, что скорость сходимости итерационного процесса решения задачи будет больше, чем в случае «дифференциального» метода КНК.

Также отметим, что в [13] было показано, что включение аппроксимации (13) интеграла (3) в переопределённую СЛАУ (14) вместо задания давления в одной точке существенно ускоряет сходимость итерационного процесса.

Сначала введём в рассмотрение в каждой ячейке  $\Omega_{ij}$  равномерную расчётную сетку, линии которой разбивают каждую сторону ячейки на  $N_{\text{sub}}$  отрезков,

где  $N_{\text{sub}} > 1$  — заданное количество отрезков вдоль каждой локальной координаты  $y_k$ ,  $k = 1, 2$ . Линии этой сетки разбивают ячейку  $\Omega_{ij}$  на  $N_c = N_{\text{sub}}^2$  подъячеек  $\Omega_{ij}^{(l)}$ ,  $l = 1, \dots, N_c$ . Значение  $N_{\text{sub}}$  задаётся так, чтобы величина  $N_{\text{sub}}^2$  была сравнимой с количеством  $m_b$  неизвестных коэффициентов в (8) или превосходящей  $m_b$ . Затем представление (8) приближённого решения задачи подставляется в уравнения (7), которые интегрируются по каждой подъячейке  $\Omega_{ij}^{(l)}$ :

$$\begin{aligned} & \xi \iint_{\Omega_{ij}^{(l)}} [\Delta u_m^{s+1} - (\text{Re} \cdot h)(u_1^s u_{m,y_1}^{s+1} + u_1^{s+1} u_{m,y_1}^s + u_2^s u_{m,y_2}^{s+1} + u_2^{s+1} u_{m,y_2}^s \\ & + q_{y_m}^{s+1})] dy_1 dy_2 = \xi \iint_{\Omega_{ij}^{(l)}} F_m dy_1 dy_2, \quad m = 1, 2; \quad l = 1, \dots, N_c. \end{aligned} \quad (20)$$

Заметим, что интегрирование в левой части равенства (20) легко осуществляется в аналитическом виде, потому что подынтегральное выражение имеет полиномиальный вид в соответствии с табл. 1.

В вышеописанном варианте метода КНК интегрируются только линеаризованные дифференциальные уравнения с частными производными с целью получения уравнений коллокаций, но при этом условия согласования используются в дифференциальной форме, как и в «дифференциальном» методе КНК. Именно поэтому термин «интегральный», применяемый для краткости в следующем разделе при описании результатов расчётов по варианту метода КНК, описанному в данном разделе, берётся в кавычки.

### 3. Использование дифференциальной и интегральной форм уравнений коллокаций

Метод КНК является гибким численно-аналитическим методом решения краевых задач для уравнений с частными производными, и эта гибкость позволяет комбинировать в одной компьютерной программе и в одной переопределённой системе (14) уравнения коллокаций, получаемые непосредственно из линеаризованного дифференциального уравнения (7) и получаемые в результате интегрирования уравнения (7) по формуле (20).

В левой части (20) имеются, среди прочих, интегралы вида

$$b_{i,j,k} \int_{y_{m1}}^{y_{m2}} y_m dy_m, \quad m = 1, 2, \quad (21)$$

где  $y_{m1}$ ,  $y_{m2}$  — заданные пределы интегрирования,  $k = 2, 3, 5, 8, 11, 12, 14$  согласно табл. 1. Если  $|y_{m1}| = |y_{m2}|$ , то интеграл (21) равен нулю, и в соответствующей строке матрицы  $A_{i,j}$  системы (14) теряется информация об одном или нескольких коэффициентах  $b_{i,j,k}$ . Следовательно, нужно подбирать подъячейки в ячейке  $\Omega_{ij}$  так, чтобы в каждой подъячейке выполнялось условие  $|y_{m1}| \neq |y_{m2}|$  хотя бы при одном  $m$ . Если оставить в матрице  $A_{i,j}$  часть или все уравнения коллокаций, получаемые из УЧП (7), то можно добавить, с целью повышения

точности расчёта по методу КНК, несколько уравнений коллокаций вида (20). При этом можно брать в (20) особенно простые пределы интегрирования, например,  $y_{m1} = 0$ ,  $y_{m2} = 1$  или  $y_{m1} = -1$ ,  $y_{m2} = 0$ . В этих случаях выражения для интегралов (20) существенно упрощаются, за счёт чего можно добиться некоторого снижения машинного времени, необходимого для решения конкретной задачи.

Пусть  $N_{c,dif}$  — количество уравнений коллокаций, полученных из дифференциальных уравнений (7), и  $N_{c,int}$  — количество уравнений коллокаций в интегральной форме (20). Можно по-разному задавать величины  $N_{c,dif}$  и  $N_{c,int}$ , нужно только следить за тем, чтобы величина  $N_c = N_{c,dif} + N_{c,int}$  была сравнимой с количеством неизвестных коэффициентов в (8) или превосходила его.

В следующем разделе приводится пример расчёта, когда в системе (14) использовались дифференциальные уравнения коллокаций и к ним ещё добавлялись четыре уравнения коллокаций в интегральной форме (20). При этом интегрирование в (20) при  $m = 1$  выполнялось по следующим двум подъячейкам  $\Omega_{ij}^{(1)}$  и  $\Omega_{ij}^{(2)}$  ячейки  $\Omega_{ij}$ :

$$\Omega_{ij}^{(1)} = \{(y_1, y_2) | -1 \leq y_1 \leq 0, -1 \leq y_2 \leq 1\}, \Omega_{ij}^{(2)} = \{(y_1, y_2) | 0 \leq y_1 \leq 1, -1 \leq y_2 \leq 1\}.$$

При  $m = 2$  интегрирование выполнялось в (20) при другом разбиении ячейки  $\Omega_{ij}$  на две подъячейки:

$$\Omega_{ij}^{(3)} = \{(y_1, y_2) | -1 \leq y_1 \leq 1, -1 \leq y_2 \leq 0\}, \Omega_{ij}^{(4)} = \{(y_1, y_2) | -1 \leq y_1 \leq 1, 0 \leq y_2 \leq 1\}.$$

$$\text{Очевидно, } \Omega_{ij} = \Omega_{ij}^{(1)} \cup \Omega_{ij}^{(2)} = \Omega_{ij}^{(3)} \cup \Omega_{ij}^{(4)}.$$

## 4. Результаты численных расчётов

### 4.1. Тестирование

Рассмотрим следующее точное решение уравнений Навье-Стокса (1) [24]:

$$\begin{aligned} u_1 &= \frac{-2(1+x_1)}{(1+x_1)^2 + (1+x_2)^2}, \quad u_2 = \frac{2(1+x_1)}{(1+x_1)^2 + (1+x_2)^2}, \\ p &= -\frac{2}{(1+x_1)^2 + (1+x_2)^2}, \quad 0 \leq x_1, x_2 \leq 1. \end{aligned} \quad (22)$$

Заметим, что функции  $u_1(x_1, x_2)$  и  $u_2(x_1, x_2)$  описывают поле скорости с нулевой дивергенцией. Далее,

$$\begin{aligned} \int_0^1 \int_0^1 p \, dx_1 dx_2 &= 4G - \pi \ln 2 - 2i \left[ \text{Li}_2 \left( -\frac{i}{2} \right) - \text{Li}_2 \left( \frac{i}{2} \right) \right] \\ &\approx -0.46261314677281549872, \end{aligned}$$

где  $i = \sqrt{-1}$ ,  $G$  — постоянная Каталана,  $G \approx 0.91596559417721901505$ ,  $\text{Li}_2(z)$  — полилогарифмическая функция. Чтобы обеспечить выполнение равенства (3) с

погрешностью, не превышающей погрешность машинных вычислений, давление  $p$  в (3) заменялось на величину  $\bar{p} = p + 0.4626131467728155$ .

Среднеквадратичные величины погрешности решения вычислялись по формулам:

$$Err(\mathbf{u}(h)) = \left[ \frac{1}{2M^2} \sum_{i=1}^M \sum_{j=1}^M \sum_{\nu=1}^2 (u_{\nu,i,j} - u_{\nu,i,j}^{ex})^2 \right]^{\frac{1}{2}},$$

$$Err(p(h)) = \left[ \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M (p_{i,j} - p_{i,j}^{ex})^2 \right]^{\frac{1}{2}},$$

где  $M$  — количество ячеек вдоль каждого координатного направления,  $\mathbf{u}_{i,j}^{ex}$  и  $p_{i,j}^{ex}$  — вектор скорости и давление, вычисленные из точного решения (22). Величины  $\mathbf{u}_{i,j}$  и  $p_{i,j}$  обозначают численное решение, полученное по методу КНК, описанному выше, вычисленное в центре ячейки  $\Omega_{ij}$ . Порядки сходимости  $\nu_u$  и  $\nu_p$  вычисляются по известным формулам [8, 10]. Пусть  $b_{i,j,l}^s$ ,  $s = 0, 1, \dots$  — значения коэффициентов  $b_{i,j,l}$  в (8) на  $s$ -ой итерации. Использовалось следующее условие окончания итераций:  $\delta b^{n+1} < \varepsilon$ , где  $\delta b^{n+1} = \max_{i,j} (\max_{1 \leq l \leq m_b} |b_{i,j,l}^{n+1} - b_{i,j,l}^n|)$ , а  $\varepsilon < h^2$  — малая положительная величина. В дальнейшем будем называть величину  $\delta b^{n+1}$  псевдопогрешностью приближённого решения. Наряду с условием  $\delta b^{n+1} < \varepsilon$  также применялся следующий критерий для окончания итераций по нелинейности:

$$\delta \mathbf{u}^{n+1} = \|\mathbf{u}^{n+1} - \mathbf{u}^n\| < \varepsilon_2,$$

где  $\|\cdot\|$  — евклидова норма вектора,  $\varepsilon_2$  — заданная малая положительная величина.

В работе [13] введено определение степени переопределённости или недоопределённости системы (15) как величины  $\chi(A) = m_r/m_c$ , где  $m_r$  и  $m_c$  — соответственно, число строк и число неизвестных в СЛАУ с матрицей  $A$ . Было исследовано влияние величины  $\chi(A)$  на число обусловленности (16) в случае применения «дифференциального» метода КНК с  $m_b = 12$  в (8). При этом использовались граничные условия Дирихле, соответствующие аналитическому решению (22).

Опустим для краткости полученные табличные данные и лишь перечислим выводы, полученные из проведённого исследования.

1°. Скорость сходимости итерационного процесса в «дифференциальном» методе КНК существенно зависит от чисел обусловленности алгебраических систем, которые необходимо решать в каждой ячейке.

2°. В тех случаях, когда в матрицу  $A$  включаются только строки, соответствующие уравнениям коллокаций, и одна строка, соответствующая аппроксимации (13) интеграла от давления, получается очень большое число обусловленности порядка  $10^5$ , то есть матрица  $A$  является плохо обусловленной. При включении в матрицу  $A$  строк, соответствующих условиям согласования, происходит уменьшение числа обусловленности на три-пять десятичных порядков в зависимости от количества ячеек сетки, точек коллокаций и точек согласования.

Таблица 2. Погрешности  $Err(\mathbf{u}^n)$ ,  $Err(p^n)$  и их порядки сходимости  $\nu_u$ ,  $\nu_p$  на последовательности сеток,  $Re = 1000$ ,  $L = 1$ ,  $N_c = 8$ . Метод КНК<sub>12</sub>

| $M$ | $Err(\mathbf{u}^n)$ | $Err(p^n)$ | $\nu_u$ | $\nu_p$ |
|-----|---------------------|------------|---------|---------|
| 10  | 1.309e-02           | 9.754e-03  |         |         |
| 20  | 5.484e-03           | 3.664e-03  | 1.26    | 1.41    |
| 40  | 1.869e-03           | 1.135e-03  | 1.55    | 1.69    |
| 80  | 4.938e-04           | 2.870e-04  | 1.92    | 1.98    |

Таблица 3. Погрешности  $Err(\mathbf{u}^n)$ ,  $Err(p^n)$  и их порядки сходимости  $\nu_u$ ,  $\nu_p$  на последовательности сеток,  $Re = 1000$ ,  $L = 1$ ,  $N_c = 8$ . Метод КНКД<sub>15</sub>

| $M$ | $Err(\mathbf{u}^n)$ | $Err(p^n)$ | $\nu_u$ | $\nu_p$ |
|-----|---------------------|------------|---------|---------|
| 10  | 2.947e-05           | 4.731e-05  |         |         |
| 20  | 6.971e-06           | 2.168e-05  | 2.08    | 1.13    |
| 40  | 2.289e-06           | 1.046e-05  | 1.61    | 1.05    |
| 80  | 9.816e-07           | 4.950e-06  | 1.21    | 1.09    |

Таблица 4. Погрешности  $Err(\mathbf{u}^n)$ ,  $Err(p^n)$  и их порядки сходимости  $\nu_u$ ,  $\nu_p$  на последовательности сеток,  $Re = 1000$ ,  $L = 1$ ,  $N_c = 8$ . Метод КНКИ<sub>15</sub>

| $M$ | $Err(\mathbf{u}^n)$ | $Err(p^n)$ | $\nu_u$ | $\nu_p$ |
|-----|---------------------|------------|---------|---------|
| 10  | 2.910e-05           | 5.437e-05  |         |         |
| 20  | 6.807e-06           | 2.249e-05  | 2.10    | 1.27    |
| 40  | 2.325e-06           | 1.076e-05  | 1.55    | 1.06    |
| 80  | 9.877e-07           | 4.893e-06  | 1.24    | 1.14    |

Именно такое значительное снижение числа обусловленности обеспечивает работоспособность метода КНК при решении краевых задач для уравнений с частными производными.

3°. Число обусловленности в граничных ячейках всегда выше, чем в случае внутренних ячеек.

4°. При  $N_c = 4$  и  $N_{\text{mat}} = 1$  существенно замедляется скорость сходимости метода КНК по сравнению со случаем  $N_c = 8$ ,  $N_{\text{mat}} = 2$ .

5°. Полученные данные являются практическим доказательством того, что при использовании переопределённой системы в приближённой задаче ( $\chi(A) > 1$ ) соответствующая СЛАУ приближённой задачи получается лучше обусловленной, чем когда СЛАУ при этом не является переопределённой. Это важное преимущество метода коллокации и наименьших квадратов перед методом коллокации, в котором не используется переопределённое СЛАУ в приближённой задаче.

Для изучения свойств сходимости и точности представленных выше вариантов метода КНК были проведены многочисленные вычислительные эксперименты с использованием аналитического решения (22). Были даны следующие названия различным вариантам метода КНК: КНКД<sub>12</sub> — «дифференциальный» метод КНК [12, 13], использующий 12 базисных векторов, то есть  $m_b = 12$  в (8); КНКД<sub>15</sub> — «дифференциальный» метод КНК с  $m_b = 15$  в (8); КНКДИ<sub>15,  $N_c$</sub>  — метод КНК с  $m_b = 15$ , в котором используются  $N_c$  уравнений коллокаций, полученных из дифференциальных уравнений (7), и четыре уравнения коллокаций в интегральной форме (20); наконец, КНКИ<sub>15</sub> — «интегральный» метод КНК с  $m_b = 15$  в (8).

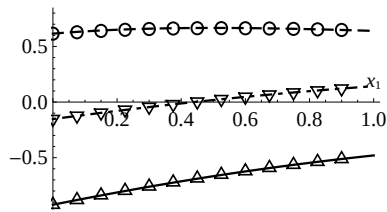


Рис. 1. Сравнение профилей приближенного и точного решений при  $x_2 = L/2$ . Сетка  $40 \times 40$  ячеек

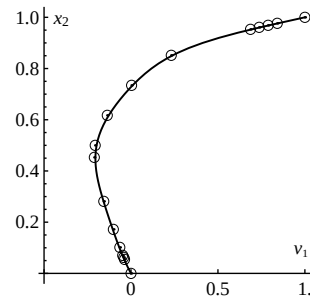


Рис. 2. Решение эталонной задачи методом КНКДИ<sub>15,4</sub> при  $Re = 100$  на сетке из  $40 \times 40$  ячеек: профиль составляющей скорости  $v_1$  вдоль центральной линии каверны  $x_1 = 0.5$  (сплошная линия); (o o o) Ghia и др. [25]

В табл. 2 – 4 представлены результаты численных экспериментов, в которых использовались два из описанных выше способов ускорения сходимости: двухпараметрический предобуславливатель и метод подпространств Крылова. В этих расчётах были взяты число Рейнольдса  $Re = 1000$ ,  $L = 1$  в (4) и  $N_c = 8$  в случае методов КНКД<sub>12</sub> и КНКД<sub>15</sub>; в случае метода КНКИ<sub>15</sub> использовали значение  $N_c = N_{sub}^2 = 16$ . Критерием окончания расчётов по вариантам метода КНК было выполнение неравенства  $\delta b^n < 10^{-10}$ . Из сравнения таблиц 2 и 3 видно, что увеличение количества базисных вектор-функций с 12 до 15 уменьшает ошибки в давлении и скорости в численном решении на два-три десятичных порядка.

На рис. 1 приведены профили компонент приближённого решения, полученного методом КНКД<sub>15</sub>, и точного решения. Компоненты  $v_1$ ,  $v_2$  и  $p$  приближенного решения нарисованы символами  $\triangle$ ,  $\circ$  и  $\nabla$ , соответствующие компоненты точного решения — сплошными, штриховыми и штрих-пунктирными линиями. Здесь видно хорошее согласие между численными результатами и аналитическим решением.

#### 4.2. Течение в квадратной каверне с движущейся крышкой

В рассматриваемой задаче расчётная область — каверна — квадрат (4) со стороной  $L = 1$ , начало координат находится в её левом нижнем углу. Верхняя крышка каверны движется в безразмерных величинах с единичной скоростью в положительном направлении оси  $Ox_1$ . Остальные стороны каверны (4) покоятся. На всех сторонах заданы условия прилипания:  $v_1 = 1$ ,  $v_2 = 0$  при  $x_2 = 1$  и  $v_m = 0$ ,  $m = 1, 2$  на остальных сторонах.

Течение в каверне с движущейся крышкой имеет сингулярности в верхних углах области. Их влияние на точность численного решения усиливается с увеличением числа Рейнольдса. Поэтому при больших числах Рейнольдса для получения более точного решения необходимо применять приём исключения особенностей решения и/или адаптивные сетки с более мелкими ячейками в окрестности сингулярностей. Здесь использовали только равномерные сетки.

Было проведено сравнение точности различных вариантов метода КНК, опи-

Таблица 5. Погрешности  $\delta u_1$ , полученные при использовании различных вариантов метода КНК,  $Re = 100$ 

| Метод                 | $\xi$ | $\eta$ | $K_{mgr}$ | $k$ | $N_{it}$ | CPU time, s. | $\delta u_1$ |
|-----------------------|-------|--------|-----------|-----|----------|--------------|--------------|
| КНКД <sub>12</sub>    | 2.0   | 3.5    | 4         | 8   | 2336     | 53.59        | 1.726e-02    |
| КНКД <sub>15</sub>    | 2.0   | 3.5    | 4         | 8   | 2540     | 99.59        | 1.150e-02    |
| КНКИ <sub>15</sub>    | 0.25  | 1.75   | 4         | 9   | 1700     | 110.38       | 8.521e-03    |
| КНКДИ <sub>15,4</sub> | 2.0   | 3.5    | 4         | 8   | 3257     | 84.66        | 8.443e-03    |
| КНКДИ <sub>15,8</sub> | 2.0   | 3.5    | 4         | 8   | 2697     | 78.89        | 1.005e-02    |

санных выше, в случае числа Рейнольдса  $Re = 100$ . Для сравнения использовалось численное решение из работы [25]. Введём в рассмотрение погрешность  $\delta u_1 = \max_j |u_{1,Ghia}(0.5, x_{2j}) - u_{1,CLR}(0.5, x_{2j})|$ , где координаты  $x_{2j}$  были взяты из [25]. Во всех расчётах, результаты которых приведены в табл. 5, использовалась комбинация всех трёх способов ускорения сходимости итераций, кратко описанных в разделе 1. Оптимальные значения параметров  $\xi$ ,  $\eta$ , входящих в двухпараметрический предобуславливатель, были найдены с помощью алгоритма [12].  $N_{it}$  — количество итераций, необходимых для выполнения неравенства  $\delta b^n < 10^{-9}$ . Значение  $k$  — количество невязок, применявшихся в методе Крылова (см. уравнение (17)). В многосеточном комплексе применялись сетки из  $5 \times 5$ ,  $10 \times 10$ ,  $20 \times 20$  и  $40 \times 40$  ячеек;  $K_{mgr}$  — количество последовательно использовавшихся сеток в многосеточном комплексе. Из табл. 5 видно, что метод КНКДИ<sub>15,4</sub> обеспечивает наилучшую точность среди четырёх рассмотренных вариантов метода КНК (см. также рис. 2).

## Заключение

Представлены новые варианты метода КНК. В численных экспериментах показано, что предложенные варианты метода КНК позволяют получать более точные численные решения, чем «дифференциальный» вариант КНКД<sub>12</sub> метода КНК.

Значительное (на два-три десятичных порядка) увеличение точности результатов, получаемых по методу КНК при небольшом (на 25%) увеличении числа базисных вектор-функций указывает на целесообразность (и осуществимость) обобщения этого приёма на случаи применения вариантов метода КНК для численного решения краевых задач для трёхмерных уравнений Навье-Стокса, так как тогда можно будет использовать для достижения заданной точности сетки с меньшим числом ячеек, чем в случае базисов меньшего размера.

Следует также отметить, что представленные варианты метода КНК очень хорошо подходят для распараллеливания, так как вычисления в каждой ячейке пространственной сетки могут выполняться на каждой итерации независимо от других ячеек. Можно разбить расчётную область на любое количество подобластей, равное числу доступных процессоров.



## ЛИТЕРАТУРА

1. Li K., Li Q. Three-dimensional gravity-jitter induced melt flow and solidification in magnetic fields // J. Thermophys. Heat Transfer. 2003. V. 17, N. 4. P. 498–508.
2. Bailly O., Buchou C., Floch A., Sainsaulieu L. Simulation of the intake and compression strokes of a motored 4-valve SI engine with a finite element code // Oil & Gas Science and Technol. 1999. V. 54. P. 161–168.
3. Плясунова А.В., Слепцов А.Г. Коллокационно-сеточный метод решения нелинейных параболических уравнений на подвижных сетках // Моделирование в механике. 1987. Т. 18, № 4. С. 116–137.
4. Семин Л.Г., Слепцов А.Г., Шапеев В.П. Метод коллокаций-наименьших квадратов для уравнений Стокса // Вычисл. технологии. 1996. Т. 1, № 2. С. 90–98.
5. Semin L., Shapeev V. Constructing the numerical method for Navier–Stokes equations using computer algebra system // Proc. 8th Internat. Workshop on Computer Algebra in Scientific Computing. LNCS. 2005. V. 3718. P. 367–378.
6. Исаев В.И., Шапеев В.П. Варианты метода коллокаций и наименьших квадратов повышенной точности для численного решения уравнений Навье–Стокса // Ж. вычисл. матем. и матем. физ. 2010. Т. 50, № 10. С. 1758–1770.
7. Shapeev V.P., Isaev V.I., Idimeshev S.V. The collocations and least squares method: application to numerical solution of the Navier–Stokes equations // CD-ROM Proc. 6th ECCOMAS, Sept. 2012, Vienna Univ. of Tech. 2012. ISBN: 978-3-9502481-9-7.
8. Shapeev V.P., Vorozhtsov E.V. Symbolic-numeric implementation of the method of collocations and least squares for 3D Navier–Stokes equations // Proc. 14th Internat. Workshop on Computer Algebra in Scientific Computing. LNCS. 2012. V. 7442. P. 321–333.
9. Шапеев В.П., Ворожцов Е.В., Исаев В.И., Идимешев С.В. Метод коллокаций и наименьших невязок для трёхмерных уравнений Навье–Стокса // Вычислит. методы и программирование. 2013. Т. 14. С. 306–322.
10. Shapeev V.P., Vorozhtsov E.V. CAS application to the construction of the collocations and least residuals method for the solution of 3D Navier–Stokes equations // Proc. 15th Internat. Workshop on Computer Algebra in Scientific Computing. LNCS. 2013. V. 8136. P. 381–392.
11. Shapeev V. Collocation and least residuals method and its applications // EPJ Web of Conferences. 2016. V. 108, N. 01009. DOI: 10.1051/epjconf/201610801009.
12. Shapeev V.P., Vorozhtsov E.V. Symbolic-numeric optimization and realization of the method of collocations and least residuals for solving the Navier–Stokes equations // Proc. 18th Internat. Workshop on Computer Algebra in Scientific Computing. LNCS. 2016. V. 9890. P. 473–488.
13. Ворожцов Е.В., Шапеев В.П. О комбинировании способов ускорения сходимости итерационных процессов при численном решении уравнений Навье–Стокса // Вычисл. методы и программирование. 2017. Т. 18. С. 80–102.
14. Shapeev A.V., Lin P. An asymptotic fitting finite element method with exponential mesh refinement for accurate computation of corner eddies in viscous flows // SIAM J. Sci. Comput. 2009. V. 31. P. 1874–1900.
15. Botella O., Peyret R. Benchmark spectral results on the lid-driven cavity flow // Comput. Fluids. 1998. V. 27. P. 421–433.
16. Carey G.G., Cheung Y.K., Lau S.L. Mixed operator problems using least squares finite element collocation // Comput. Meth. Appl. Mech. Engng. 1980. V. 22. P. 121–130.

17. Jiang B.N. The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics. Berlin: Springer, 1998. 418 p.
18. Bochev P.B., Gunzburger M.D. Least-Squares Finite Element Methods. New York: Springer-Verlag, 2009.
19. Ranjan R., Chronopoulos A.T., Feng Y. Computational algorithms for solving spectral/hp stabilized incompressible flow problems // J. Math. Res. 2016. V. 8, N. 4. P. 21–39.
20. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения. М. : Мир, 2001. 430 с.
21. Крылов А.Н. О численном решении уравнения, которым в технических вопросах определяются частоты малых колебаний материальных систем // Изв. АН СССР, Отд. матем. и естеств. наук. 1931. № 4. С. 491–539.
22. Федоренко Р.П. О скорости сходимости одного итерационного процесса // Ж. вычисл. матем. и матем. физ. 1964. Т. 4, № 3. С. 559–564.
23. Wesseling P. An Introduction to Multigrid Methods. Chichester: John Wiley & Sons, 1992. 284 p.
24. Chiu P.H., Sheu T.W.H., Lin R.K. An effective explicit pressure gradient scheme implemented in the two-level non-staggered grids for incompressible Navier–Stokes equations // J. Comput. Phys. 2008. V. 227. P. 4018–4037.
25. Ghia U., Ghia K.N., Shin C.T. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method // J. Comput. Phys. 1982. V. 48. P. 387–411.

#### APPLICATION OF THE INTEGRAL FORM OF COLLOCATION EQUATIONS AND DIFFERENTIAL MATCHING CONDITIONS IN THE METHOD OF COLLOCATIONS AND LEAST SQUARES

**E.V. Vorozhtsov<sup>1</sup>**

Leading Research Scientist, Dr.Sc. (Phis.-Math.), e-mail: vorozh@itam.nsc.ru

**V.P. Shapeev<sup>1,2</sup>**

Professor, Dr.Sc. (Phis.-Math.), e-mail: vshapeev@ngs.ru

<sup>1</sup>Khristianovich Institute of Theoretical and Applied mechanics SB RAS, Novosibirsk

<sup>2</sup>Faculty of mechanics and mathematics, Novosibirsk National Research University

**Abstract.** To increase the accuracy of computations by the method of collocations and least squares (CLS) it is proposed to increase the number of degrees of freedom with the aid of the following two techniques: an increase in the number of basis vectors and the integration of the linearized partial differential equations (PDEs) over the subcells of each cell of a spatial computational grid. It is shown that the proposed new versions of the CLS method possess a higher accuracy than the previous versions of this method. Furthermore, the version of the CLS method, which employs the integral form of collocation equations, needs a lesser number of iterations for its convergence than the “differential” CLS method.

**Keywords:** method of collocations and least squares, preconditioning, Krylov subspaces, multigrid, collocation of integral relations, Navier-Stokes equations.

*Дата поступления в редакцию: 28.04.2017*

## **ПОСТРОЕНИЕ МЕТАМОРФИЗМОВ РАСТРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ РЕШЕНИЯ УРАВНЕНИЙ ЭЙЛЕРА-ПУАНКАРЕ**

**С.Н. Чуканов<sup>1</sup>**

профессор, д.т.н., ведущий научный сотрудник, e-mail: ch\_sn@mail.ru

**С.В. Лейхтер<sup>2</sup>**

аспирант, e-mail: leykhter@mail.ru

<sup>1</sup>ФГБУН Институт математики им. С.Л. Соболева СО РАН, Омский филиал

<sup>2</sup>ФГБОУ ВО Сибирский государственный автомобильно-дорожный университет  
(СибАДИ)

**Аннотация.** В работе рассмотрена задача сравнения двух растровых изображений — исходного и целевого, представленных соответствующим каждому множеством чёрно-белых пикселей. Задача решается путём нахождения диффеоморфизма, который позволит совместить деформируемое изображение образа с изображением шаблона. В основе решения лежит метод построения функционала, характеризующего эволюцию диффеоморфизмов изображения от его начального состояния до конечного и «штраф» за отклонение траекторий движения точек изображения от требуемых. Разработан алгоритм решения уравнения диффеоморфизма, основанный на оптимизации (минимизации) построенного функционала методом градиентного спуска. Показан переход от метаморфизма точечных множеств к метаморфизму изображений, благодаря чему становится возможным оптимальный метаморфизм в случаях, когда отсутствует поточечное соответствие между исходными и целевыми объектами. Предложенные в работе алгоритмы могут использоваться в биометрических системах, системах классификации изображений и объектов, системах машинного зрения, при распознавании образов и объектов, системах трекинга.

**Ключевые слова:** распознавание образов, уравнения Эйлера-Пуанкаре, диффеоморфизм, метаморфизм изображений.

### **Введение**

При сопоставлении изображений сравниваемых объектов цель состоит в нахождении оптимального преобразования одного изображения в другое. На основе этого принципа был разработан метод диффеоморфного сопоставления образов и определения римановой метрической структуры на пространствах деформируемых объектов [3]. Этот метод решает проблему регистрации изображений путём решения вариационной задачи нахождения минимума

$(d(\text{id}, g)^2 + \|g \cdot n_{\text{temp}} - n_{\text{targ}}\|)$  по всем элементам группы диффеоморфизмов  $g$ , где исходное изображение (образ)  $n_{\text{temp}}$  и целевое (шаблон)  $n_{\text{targ}}$  — сравниваемые объекты,  $g \cdot n$  — действие элемента группы диффеоморфизмов  $g$  на изображение  $n$  и  $d$  — риманово расстояние на диффеоморфизмах, однако, член  $\|g \cdot n_{\text{temp}} - n_{\text{targ}}\|$  нарушает метрические аспекты группы диффеоморфизмов. С целью разработки метрического подхода к проблеме сопоставления шаблонов были введены метаморфизмы изображений [7], которые обеспечивают метрическую структуру.

Использование теоретико-групповых методов анализа изображений является инструментом исследований в компьютерной визуализации. Аналитические и вычислительные свойства низкоразмерных матричных групп Ли составляют основу компьютерной графики. В работе М. Miller and L. Younes [6] представления изображений с помощью группы матриц конечной размерности заменяются их бесконечномерным аналогом — группами диффеоморфизмов. Деформируемый шаблон изображения превращается в метрическое пространство с метрическим расстоянием между элементами путём построения кривых через пространство диффеоморфизмов, соединяющих их. Длина кривой становится основой для построения метрического расстояния, которое соответствует кривым геодезической кратчайшей длины, что приводит к вариационной задаче, описывающей геодезические потоки между элементами изображения на орбите, с решением уравнений Эйлера-Лагранжа, определяющих оптимальный поток диффеоморфизмов. Одной из целей настоящей работы является разработка бесконечномерных аналогов для исследования многомерных форм изображений через лиевы группы диффеоморфизмов.

В работе рассмотрена задача сравнения двух растровых изображений — исходного и целевого, представленных соответствующими множествами чёрно-белых пикселей. Задача решается путём нахождения диффеоморфизма, который позволит совместить деформируемое изображение образа с изображением шаблона. В основе решения лежит метод построения функционала, характеризующего эволюцию диффеоморфизмов изображения от его начального состояния до конечного и «штраф» за отклонение траекторий движения точек изображения от требуемых. При использовании вариационного исчисления задача приводится к обыкновенным дифференциальным уравнениям первого порядка Эйлера-Пуанкаре. Разработан алгоритм решения уравнения диффеоморфизма, основанный на оптимизации (минимизации) построенного функционала методом градиентного спуска. Показан переход от метаморфизма точечных множеств к метаморфизму изображений, благодаря чему становится возможным оптимальный метаморфизм в случаях, когда отсутствует поточечное соответствие между исходными и целевыми объектами. Предложенные в работе алгоритмы могут использоваться в биометрических системах, системах классификации изображений и объектов, системах машинного зрения, при распознавании образов и объектов, системах трекинга.

Отличительной особенностью предлагаемого метода от существующих методов метаморфизма изображений является то, что функционал, характеризующий диффеоморфизм изображений, является оптимальным и уникальным.

## 1. Уравнения Эйлера-Пуанкаре

Рассмотрим 0-мерные многообразия совокупности точек, 1-мерные многообразия кривых и многообразия изображений, которые могут быть представлены скалярами, векторами и матрицами. Пусть на многообразии  $\mathcal{M}$  действуют элементы  $g$ ;  $g_0 = g(t=0) = \text{id}$  лиевой группы диффеоморфизмов  $G$ , параметризованных временем  $t$ . Метаморфизм определяет пару кривых: шаблон метаморфизма  $\eta \in \mathcal{M}$  и образ метаморфизма  $n \in \mathcal{M}$ , который определяется орбитой действия элементов группы  $g$ :  $n = g \cdot \eta \in \mathcal{M}$ . Если  $\eta = \text{const}$ , — метаморфизм является диффеоморфизмом.

Будем считать, что лиева группа — это группа, которая является конечно-мерным вещественным гладким многообразием, в котором групповые операции являются гладкими отображениями. Дифференцируемое отображение  $\mathcal{M} \rightarrow \mathcal{N}$  для многообразий  $\mathcal{M}, \mathcal{N}$  будет диффеоморфизмом, если оно является биекцией и его обратное отображение  $\mathcal{N} \rightarrow \mathcal{M}$  является дифференцируемым. Лиева группа диффеоморфизмов  $\text{Diff}(\mathcal{M})$  дифференцируемого многообразия  $\mathcal{M}$  является группой всех диффеоморфизмов  $\mathcal{M}$  к самой себе.

Метрическая длина близости двух многообразий (совокупности точек, кривых или изображений) определяется самой короткой длиной потока, который переносит систему координат одного многообразия формы в систему координат другого. Для сравнения двух изображений, которые являются многообразиями, найдём оптимальную деформацию, которая переводит одно изображение в другое. Для этого сформируем минимизируемый функционал  $\int_0^1 L(g, \dot{g}, \eta, \dot{\eta}) dt$ , с функцией Лагранжа  $L$  и с фиксированными граничными условиями  $n_0 = \eta_0 \in \mathcal{M}$ ,  $n_1 = g_1 \cdot \eta_1 \in \mathcal{M}$ . Будем считать, что функция  $L$  инвариантна при правом действии элемента группы  $h \in G$  [2, 9]:  $(g, \eta)h = (gh, h^{-1}\eta)$ . Для метаморфизма  $(g, \eta)$  введём скорость изменения элемента группы  $g$ , которая является элементом соответствующей лиевой алгебры  $\mathfrak{g}$ :  $u = \dot{g}g^{-1} \in \mathfrak{g}$ ; и  $n = g\eta$ ;  $\nu = g\dot{\eta}$ , откуда  $\nu = \dot{n} - un$  (действие лиевой алгебры обозначается конкатенацией с левой стороны). Тогда лагранжиан может быть редуцирован  $L(g, \dot{g}, \eta, \dot{\eta}) = l(u, n, \nu)$ . Сравнение образов основано на решении вариационной задачи  $\delta \int l(u, n, \nu) dt = 0$ , для диффеоморфизмов  $g \in G$ .

Из условия оптимизации действия  $S = \int_0^1 l(u, n, \nu) dt$  по отношению к вариации  $\delta u$  и  $\delta n = \delta(g\eta) = \omega$  при фиксированных  $n_0, n_1$  определим уравнения Эйлера-Пуанкаре [4, 5]. Найдём производную по времени  $d/dt(\delta n)$ :  $d/dt(\delta n) = \dot{\omega} = \delta\nu + u\omega + (\delta u)n$ . Вариационное условие оптимизации  $\delta S = 0$  приводит к уравнению

$$\int_0^1 (\langle \delta l / \delta u, \delta u \rangle + \langle \delta l / \delta n, \omega \rangle + \langle \delta l / \delta \nu, \dot{\omega} - u\omega - (\delta u)n \rangle) dt = 0. \quad (1)$$

Равенство нулю членов с  $\delta u$  приводят к уравнению

$$\delta l / \delta u + \delta l / \delta \nu \diamond n = 0, \quad (2)$$

здесь оператор  $\diamond$  определяется из соотношения  $\langle v \diamond a, \xi \rangle = \langle a, \xi v \rangle$ ;  $\forall \xi \in \mathfrak{g}; v \in \in V; a \in V^*$ . После интегрирования (1) по частям получим

$$\partial / \partial t \left( \delta l / \delta v \right) + u \star \delta l / \delta v - \delta l / \delta n = 0, \quad (3)$$

где  $\star$  определяется из соотношения  $\langle u \star \delta l / \delta v, \omega \rangle = \langle \delta l / \delta v, u \omega \rangle$ . Для построения полной системы уравнений добавим соотношение

$$\dot{n} = v + un. \quad (4)$$

Диффеоморфизмы  $g(x) \in G; x \in M$ , которые могут быть представлены в форме потоков обыкновенных дифференциальных уравнений, эволюционирующих во времени  $t \in [0, 1]$  с векторным полем  $u(\cdot)$  [5]

$$dg(x)/dt = u(g(x)); g_{t=0}(x) = x. \quad (5)$$

Запишем скалярное произведение в пространстве  $\mathfrak{g}$  в форме  $\langle u, v \rangle_{\mathfrak{g}} = \int_M (L_{\mathfrak{g}} u)^* v; \|u\|_{\mathfrak{g}}^2 = \langle u, u \rangle_{\mathfrak{g}}$ , где  $L_{\mathfrak{g}}$  — оператор дуальности, который отображает элементы лиевой алгебры  $\mathfrak{g}$  на элементы лиевой коалгебры  $\mathfrak{g}^*$ :  $L_{\mathfrak{g}} : \mathfrak{g} \rightarrow \mathfrak{g}^*$ . В случае механической интерпретации величина  $L_{\mathfrak{g}} u$  имеет смысл вектора импульса,  $u$  — вектора скорости,  $L_{\mathfrak{g}}$  — тензора инерции. Введём обратный оператор  $K$ , которой формально представим в виде  $K = L_{\mathfrak{g}}^{-1}$ .

Для элементов группы  $g_t \in G; t \in [0, \dots, 1]$  существуют скорости изменения  $g_t$ :  $u = dg_t / dt g_t^{-1} \in \mathfrak{g}$ , которые минимизируют функционал  $S = \int_0^1 l(u) dt$  с лагранжианом

$$l = \|u\|_{\mathfrak{g}}^2 + \sigma^{-2} |v|^2 \quad (6)$$

на траектории, соединяющей элементы группы  $g_0 = g|_{t=0}$  и  $g_1 = g|_{t=1}$ ; здесь второе слагаемое накладывает штраф за отклонение  $|v|^2 = |\dot{n} - un|^2$ , обратно пропорциональный величине  $\sigma^2$ .

Выражение для импульса запишем в виде  $p = L_{\mathfrak{g}} u$ ; тогда обратное выражение  $u = L_{\mathfrak{g}}^{-1} p = Kp$ , или  $u(x) = \int_{\Omega} K(x, y) p(y) dy$ .

Для оператора  $L = \text{id} - a \nabla^2$  в  $\mathbb{R}^2$  — обратный оператор  $K = L^{-1}$  аппроксимируем функцией Гаусса [8]

$$K(x, y) = \beta e^{-\alpha^{-1} |x-y|^2}. \quad (7)$$

Уравнения эволюции диффеоморфизмов Эйлера-Пуанкаре можно получить решением уравнений вариационной задачи с функционалом (6) [1]:

$$\begin{aligned} \dot{g} &= u(g); u = L_{\mathfrak{g}}^{-1} p = Kp; \\ \dot{p} &= -\text{ad}_u^* p. \end{aligned} \quad (8)$$

## 2. Метаморфизм точечных множеств

Пусть заданы два множества  $n_0 = (x_1, \dots, x_N), n_1 = (y_1, \dots, y_N)$  помеченных точек в  $M$ . Поставим задачу нахождения такого минимального диффеоморфизма  $g : M \rightarrow M$ , что  $g(x_i) \simeq y_i; i = 1, \dots, N$  (неточное соответствие). Множество диффеоморфизмов  $\text{Diff}(M)$  определяют структуру группы.

Для точечных множеств  $p(y) = \sum_{i=1}^N p_i \cdot \delta(y - q_i)$ , откуда получим

$$u(x) = \sum_{i=1}^N K(x, q_i) p_i. \quad (9)$$

Пусть пространство  $M$  содержит  $N$  объектов (точек), которые подлежат деформации. Рассмотрим действие  $S = \int_0^1 l(u, \nu) dt$  с лагранжианом

$$l(u, \nu) = \|u\|_{\mathfrak{g}}^2 + 1/\sigma^2 \sum_{k=1}^N |\nu_k|^2 = \|u\|_{\mathfrak{g}}^2 + 1/\sigma^2 \sum_{k=1}^N |\dot{q}_k - u(q_k)|^2, \quad (10)$$

который не зависит от  $n$ . Тогда  $\delta l / \delta u = 2L_{\mathfrak{g}}u$ , и:  $\delta l / \delta \nu = 2/\sigma^2 (\nu_1, \dots, \nu_N)$ . Из соотношений (2, 3, 4)  $\delta l / \delta u + \delta l / \delta \nu \diamond n = 0$ , имеем  $\delta l / \delta \nu \diamond n = -2/\sigma^2 \sum_{k=1}^N \nu_k \cdot \delta(q - q_k)$ , поэтому

$$L_{\mathfrak{g}}u = \sigma^{-2} \sum_{k=1}^N \nu_k \cdot \delta(q - q_k), \quad (11)$$

откуда  $u(q) = \sum_{k=1}^N K(q, q_k) p_k$ , где  $p_k = \sigma^{-2} \nu_k$ .

Из уравнения (3) получим

$$\dot{p}_k + Du(q_k)^T p_k = \dot{p}_k + \sum_{l=1}^N \nabla_1 K(q_k, q_l) p_k^T p_l = 0; k = 1, \dots, N, \quad (12)$$

где  $\nabla_1 K$  представляет собой градиент функции  $(q_k, q_l) \rightarrow K(q_k, q_l)$  по отношению к  $q_k$ . Если для оператора  $L_{\mathfrak{g}}$  выбрать функцию Грина в виде  $K(q_k, q_l) = \beta e^{-\alpha^{-1}|q_k - q_l|^2}$ , то  $\nabla_1 K(q_k, q_l) = -2\alpha^{-1}\beta(q_k - q_l)e^{-\alpha^{-1}|q_k - q_l|^2}$ .

Из уравнения (4):  $\dot{n} = \nu + un$ , и обозначения  $n = (q_1, \dots, q_N)$ , получим  $\dot{q}_k = v_k + u(q_k) = u(q_k) + \sigma^2 p_k; k = 1, \dots, N$ .

Перепишем систему (2, 3, 4) для точечных множеств в виде

$$\begin{aligned} \dot{q}_k &= u(q_k) + \sigma^2 p_k; u(q) = \sum_{k=1}^N K(q, q_k) p_k; k = 1, \dots, N; \\ \dot{p}_k + \sum_{l=1}^N \nabla_1 K(q_k, q_l) p_k^T p_l &= 0. \end{aligned} \quad (13)$$

Решение системы (13) сводится к решению двухточечной краевой задачи, так как множество  $n = (q_1, \dots, q_N)$  задано в начальный момент времени, а множество  $(p_1, \dots, p_N)$  — в конечный момент. Рассмотрим применение метода градиентного спуска в задаче нахождения метаморфизма  $g_{10} \cdot q_k(0) \rightarrow q_k(1)$  при минимизации функционала  $1/\sigma^2 \sum_{k=1}^N [q_k(1) - g_{10} \cdot q_k(0)]^2 \rightarrow \min$  итерационным выбором начальных значений вектора  $p(0) = (p_1(0), \dots, p_N(0))$ :  $p(0)^{it+1} = p(0)^{it} - \gamma \cdot \left( \partial S / \partial p(0) \big|_{p(0)=p(0)^{it}} \right)^T \left( p(0)^{it} \right)$ , где  $it$  — номер итерации,  $\gamma$  — значение шага метода градиентного спуска,  $S(p(0)^{it})$  — значение действия при начальных значениях вектора  $p(0)^{it}$ .

**Пример 1.** Рассмотрим пример нахождения энергии деформации  $S$  с лагранжианом (10) на основе интегрирования дифференциальных уравнений (13) при метаморфизме квадрата в окружность, которые задаются 16 точками, равномерно расположенными на кривой (рис. 1). Точки расположены в пределах квадрата  $0 \leq x_i, y_i \leq 1$ ;  $i = 1, \dots, 16$ .

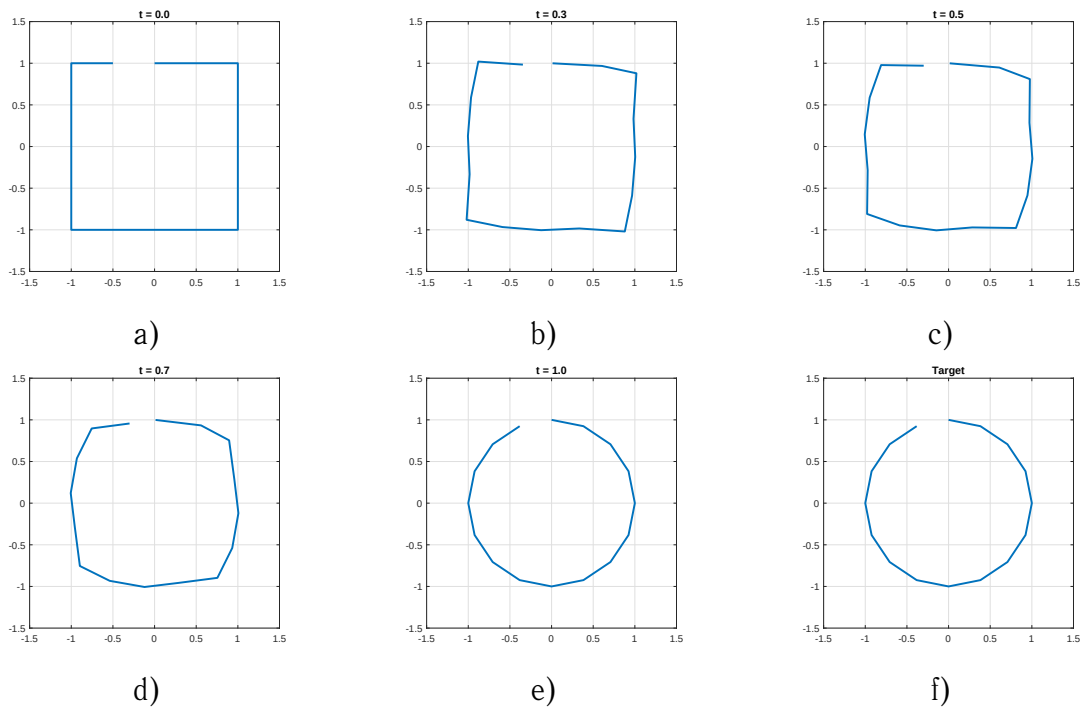


Рис. 1. Эволюции диффеоморфизма в моменты времени: а)  $t = 0,0$  (начальный образ); б)  $t = 0,3$ ; в)  $t = 0,5$ ; д)  $t = 0,7$ ; е)  $t = 1,0$ ; ф) цель (требуемый образ).

**Пример 2.** Рассмотрим пример нахождения энергии деформации на основе интегрирования дифференциальных уравнений (11) при метаморфизме образов греческих строчных символов  $\alpha, \delta, \gamma, \rho, \sigma$  (таблица 1), которые представлены 16 точками, равномерно расположенными на кривой символов; точки расположены в пределах квадрата  $0 \leq x_i, y_i \leq 1$ ;  $i = 1, \dots, 16$ . Интегрирование уравнений



проводилось при  $\alpha = 10; \sigma^2 = 0$ . Энергия вычислялась на основе определения действия  $S = \int_0^1 l(u, \nu) dt$  с лагранжианом (10).

Таблица 1. Энергии деформации при метаморфизме образов греческих строчных символов  $\alpha, \delta, \gamma, \rho, \sigma$

|          | $\alpha$ | $\delta$ | $\gamma$ | $\rho$ | $\sigma$ |
|----------|----------|----------|----------|--------|----------|
| $\alpha$ | 0        | 31,9     | 28,6     | 37,4   | 26,2     |
| $\delta$ | 31,4     | 0        | 21,6     | 39,1   | 14,9     |
| $\gamma$ | 26,3     | 21,9     | 0        | 46,7   | 15,2     |
| $\rho$   | 38,2     | 38,9     | 46,8     | 0      | 44,3     |
| $\sigma$ | 26,1     | 15,0     | 14,9     | 44,5   | 0        |

Из таблицы следует, что наименьшую энергию деформации требуют метаморфизмы  $\sigma \Leftrightarrow \delta$  и  $\sigma \Leftrightarrow \gamma$ ; наибольшую энергию деформации требуют метаморфизмы  $\rho \Leftrightarrow \sigma$  и  $\rho \Leftrightarrow \gamma$ .

### 3. Метаморфизм изображений

Рассмотрим случай, когда  $\mathcal{N}$  — пространство гладких функций (изображений)  $\Omega \rightarrow \mathbb{R}$  с действием  $(g, n) \rightarrow n \circ g^{-1}$ . Будем обозначать координаты элемента изображения (пикселя) символом  $q \in \Omega \in \mathbb{R}^d$  (в случае плоских изображений  $d = 2$ ). Простой случай метаморфизма можно получить с помощью действия:  $S = \int_0^1 l(u, \nu) dt$  с лагранжианом  $l(u, \nu) = \|u\|_{\mathfrak{g}}^2 + 1/\sigma^2 |\nu|^2$ . Если  $w \in \mathfrak{g}$  и  $n$  представляет собой изображение  $wn = -\nabla n^T w$  так, что  $(\delta^l/\delta\nu \diamond n|w) = (\delta^l/\delta\nu| \nabla n^T w)$ . Обозначим  $z = \nu/\sigma^2$ . Тогда получим уравнение  $L_{\mathfrak{g}} u_t = -z_t \nabla n$ , следовательно,  $u(\cdot) = -L_{\mathfrak{g}}^{-1}(z \nabla n) = -\int_{\Omega} K(\cdot, q) z(q) \nabla n(q) dq; q \in \Omega$ .

Так как  $u \star (\delta^l/\delta\nu)$  определяется из  $(u \star (\delta^l/\delta\nu)|\omega) = (\delta^l/\delta\nu|u\omega) = 1/\sigma^2 (\text{div}(\nu n)|\omega)$ , то получаем второе уравнение  $\dot{z}_t + \nabla \cdot (z_t \cdot u_t) = \dot{z}_t + \text{div}(z_t \cdot u_t) = 0$ .

Перепишем уравнения эволюции (2, 3, 4) в виде:

$$\begin{aligned} \dot{z}_t + \nabla \cdot (z_t \cdot u_t) &= 0; \\ \dot{n} + (\nabla n^T) \cdot u &= \sigma^2 z; \\ u(\cdot) &= -\int_{\Omega} K(\cdot, q) z(q) \nabla n(q) dq; q \in \Omega. \end{aligned} \tag{14}$$

Для дискретного двухмерного случая будем считать  $q_{ij} = (x_i, y_j)^T$ ;  $\Delta = \Delta_x = \Delta_y; dq = \Delta^2$ . Запишем выражение для  $u$  в виде  $u(q) = -\int_{\Omega} K(q, \tilde{q}) z(\tilde{q}) \nabla n(\tilde{q}) d\tilde{q} \approx \sum_j^N \sum_i^N K(q, \tilde{q}_{ij}) z(\tilde{q}_{ij}) \nabla n(\tilde{q}_{ij}) \Delta^2$ ;

$q, \tilde{q}_{ij} \in \Omega \in \mathbb{R}^2$ . Оператор RKHS  $K = L^{-1}$  аппроксимируем функцией Гаусса  $K(q_1, q_2) = \beta e^{-\alpha^{-1}|q_1 - q_2|^2} = \beta e^{-\alpha^{-1}(|x_1 - x_2|^2 + |y_1 - y_2|^2)}$ .

Решение системы (14) сводится к решению двухточечной краевой задачи, так как изображение  $n$  задано в начальный момент времени, а  $z$  — в конечный момент времени, поэтому для решения уравнений (10) применяется метод градиентного спуска (аналогично решению задачи (13)).

**Пример 3.** Рассмотрим пример метаморфизмов изображений арабских цифр при  $\sigma^2 = 10$ ,  $\beta = 0,1$ ,  $\alpha = 10$ ,  $N = 11$ . Энергия диффеоморфизмов приведена в таблице 2.

Таблица 2. Энергии диффеоморфизмов изображений арабских цифр

|   | 0      | 1      | 2      | 3      | 4      |
|---|--------|--------|--------|--------|--------|
| 0 | 0      | 0,1899 | 0,1689 | 0,0347 | 0,1963 |
| 1 | 0,3477 | 0      | 0,3375 | 0,3287 | 0,1606 |
| 2 | 0,3299 | 0,1587 | 0      | 0,2486 | 0,2062 |
| 3 | 0,0406 | 0,1884 | 0,2524 | 0      | 0,2046 |
| 4 | 0,3695 | 0,2934 | 0,4091 | 0,2044 | 0      |

## Заключение

В работе рассмотрена задача сравнения двух изображений. Для её решения предлагается нахождение диффеоморфизма, который позволит совместить изображение образа и изображение шаблона. Отличительная особенность предлагаемого метода решения заключается в том, что задача решается на основе метода построения минимизируемого функционала, характеризующего эволюцию диффеоморфного преобразования изображения от начального до конечного и штраф за отклонение траектории изображения от требуемой. При использовании вариационного исчисления задача приводится к обыкновенным дифференциальным уравнениям первого порядка Эйлера-Пуанкаре. В результате, разработан алгоритм решения уравнения диффеоморфного преобразования на основе метода градиентного спуска.

Предложенный метод сравнения изображений может быть распространен на совокупность изображений, проекции трехмерных изображений, цветные изображения и т. п. Рассмотренная задача сравнения двух изображений может быть использована при оптимальном метаморфизме изображений в случаях, когда отсутствует поточечное соответствие между исходными и целевыми объектами. Разработанные алгоритмы могут быть применены в биометрических системах, системах классификации изображений и объектов, системах машинного зрения, при распознавании образов и объектов. Например, для идентификации человека по изображению лица, отпечаткам пальцев, рисункам ладоней рук, рисункам вен ладони или пальца, подписям и т. п.

Описанный способ сравнения образов позволит добиться лучших результатов распознавания, по сравнению с известными.

## ЛИТЕРАТУРА

1. Arnold V.I., Khesin B.A. Topological methods in hydrodynamics. Springer Science & Business Media, 1999. Т. 125.
2. Chukanov S.N. Definitions of invariants for n-dimensional traced vector fields of dynamic systems // Pattern Recognition and Image Analysis. 2009. Т. 19, № 2. P. 303–305.
3. Dupuis P., Grenander U., Miller M.I. Variational problems on flows of diffeomorphisms for image matching // Quarterly of applied mathematics. 1998. P. 587–600.
4. Holm D.D., Schmäh T., Stoica C. Geometric mechanics and symmetry: from finite to infinite dimensions. Oxford University Press, 2009.
5. Holm D.D., Trounev A., Younes L. The Euler-Poincaré theory of metamorphosis // Quarterly of Applied Mathematics. 2009. Т. 67, № 4. P. 661–685.
6. Miller M.I., Younes L. Group actions, homeomorphisms, and matching: A general framework // International Journal of Computer Vision. 2001. Т. 41, № 1–2. P. 61–84.
7. Trounev A., Younes L. Metamorphoses through lie group action // Foundations of Computational Mathematics. 2005. Т. 5, № 2. P. 173–198.
8. Younes L., Arrate F., Miller M.I. Evolutions equations in computational anatomy // NeuroImage. 2009. Т. 45, № 1. P. 40–50.
9. Чуканов С.Н. Формирование инвариантов при визуализации векторных полей, определяемых интегральными кривыми динамических систем // Автометрия. 2011. Т. 47, № 2. С. 58–63.

## CONSTRUCTING METAMORPHOSIS OF PIXEL IMAGES FOR THE OBJECTS ON THE BASIS OF SOLVING EULER-POINCARÉ EQUATIONS

**S.N. Chukanov**<sup>1</sup>

Dr.Sc.(Eng.), Professor, Senior Scientist Researcher, e-mail: ch\_sn@mail.ru

**S.V. Lejhter**<sup>2</sup>

Graduate Student, e-mail: lejhter@mail.ru

<sup>1</sup>Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, Omsk branch

<sup>2</sup>Department of “Computer Information Automated Systems”, State Automobile and Highway University

**Abstract.** This work considers comparison of two images (original and target) that are represented by black and white pixels to each correspondingly. The problem can be solved by detecting diffeomorphism that would allow to match deformed and template images. Solution to the problem is based on the method of constructing a minimized functional, which characterizes evolution process of the diffeomorphic transformation

of the image from initial to the final one and “penalty charge” for the deviation of the image path from the required trajectory. An algorithm for solving the diffeomorphic transformation equation is developed on the basis of the gradient descent method. The considered problem of comparing two images can be used for constructing optimal metamorphosis of images, when there is no exact correspondence between the target image and the final image of the diffeomorphism. The designed algorithms can be used through a biometrical system, in images and subjects classification systems, machine vision systems, images and patterns recognition, tracking systems.

**Keywords:** pattern recognition, Euler-Poincare equation, diffeomorphism, metamorphosis approach.

*Дата поступления в редакцию: 26.05.2017*

## **О ФОРМАЛИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Е.А. Тюменцев**

генеральный директор, e-mail: etyumentcev@hwdtech.ru

LLC Hello World! Technologies

**Аннотация.** В книге «Мифический человеко-месяц, или Как создаются программные системы» Брукс ссылается на несколько исследований, на основании которых можно сделать вывод, что производительность труда программистов, измеряемая в количестве строк кода в единицу времени, падает по мере роста размера программного проекта. В настоящей работе описывается формализация процесса разработки программного обеспечения, как процесса редактирования исходного текста программы, с помощью которой определяется функция трудоёмкости, а также выводится достаточное условие постоянной производительности труда программистов, не зависящей от размеров проекта.

**Ключевые слова:** формализация, процесс разработки, программное обеспечение, производительность труда, Брукс, мифический человеко-месяц.

### **Введение**

В статье 2000 года [1], посвящённой двадцатипятилетию с момента выхода первого издания книги Фредерика Брукса «Мифический человеко-месяц, или Как создаются программные системы», были отмечены две важные особенности индустрии разработки программного обеспечения:

- «невиданные в истории человечества темпы смены технологий»,
- «лавинообразный рост численности персонала, занятого в отрасли».

Эти две особенности характерны для отрасли и на сегодняшний день.

Брукс в [2] ссылается на исследование [3], выполненное Наусом и Фарром в компании System Development Corporation, результаты которого вполне могут объяснить причины возникновения и даже усиления этих особенностей. Наус и Фарр измерили объём работ как функцию от количества машинных команд на 11 крупных программных проектах. Оказалось, что зависимость имеет вид степенной функции с показателем 1,5. Для наглядности, продемонстрируем результаты этого исследования в виде графика на рисунке 1.

На рисунке 1 ось ординат — объём работ, измеряемый в человеко-месяцах, ось абсцисс — количество машинных команд в тысячах, составляющих программный проект. Сплошная линия — график зависимости объёма работ от количества машинных команд, составляющих программный проект, полученный

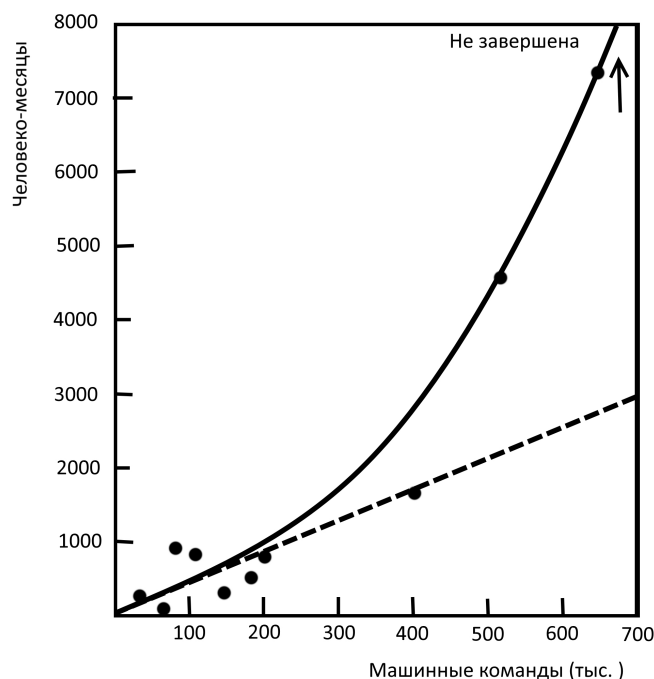


Рис. 1. График зависимости объема работ от количества машинных команд

Наусом и Фарром, — является экстраполяцией эмпирических данных. Пунктирная линия показывает, как бы выглядел график зависимости объема работ, если бы производительность труда не зависела от общего размера проекта.

Как видно из графика, производительность труда программистов может снижаться по мере роста размера программы. Опираясь на данное предположение, можно, например, объяснить следующие факты из области разработки программного обеспечения:

- частая смена технологий, выпуск новых версий, переписанных заново — начинать с нуля дешевле и быстрее, чем добавлять новый функционал в существующий продукт;
- дефицит кадров — чтобы сохранять один и тот же темп выпуска нового функционала, по мере роста размера проекта, требуется всё больше сотрудников;
- разбиение проекта на итерации — чем меньше сроки планирования, тем меньше разница между планируемой (постоянной) и фактической производительностью, тем точнее оценка сроков разработки;
- практика измерения velocity, применяемая в Agile-проектах, — это кусочно-линейная аппроксимация степенной функции (сплошная линия на рисунке 1). Velocity — это коэффициент, показывающий разницу между планируемым и фактическим объемом работ. Используется для улучшения точности оценки предстоящего объема работ: оценка, сделанная проектной командой, умножается на текущий velocity. Как правило,

velocity пересчитывается в конце каждой итерации;

- за программистами закрепились устойчивая репутация, что они срывают сроки, в том числе и потому, что прогнозы сроков даются исходя из предположения по постоянной производительности труда (пунктирная линия на рисунке 1).

Измерения, сделанные Наусом и Фарром [3] более чем 50 лет назад, были проведены только для 11 программных проектов, которые, по сегодняшним меркам, нельзя назвать крупными, поэтому нельзя утверждать, что выявленная эмпирическая закономерность справедлива хотя бы для некоторой части современных программных проектов.

Соответственно, возникают вопросы:

- Существует ли эффект падения производительности труда программиста с ростом размера проекта при использовании современных технологий?
- Если да, то можно ли его формально обосновать?
- Можно ли определить условия, при которых производительность труда не будет падать?

В настоящей статье предлагается формализация процесса разработки программного кода как процесса редактирования текста программы и выводится достаточное условие поддержания постоянной производительности труда программиста на всём протяжении проекта.

## 1. Идея

Прежде чем переходить к строгим математическим рассуждениям, опишем неформально идею.

Программа — это набор слов (который также является словом), записанных на формальном языке. Каждое слово — это последовательность лексем — символов алфавита формального языка. Когда программист вносит изменения в текст программы, то он либо добавляет новое слово, либо удаляет или редактирует существующее, добавляя или удаляя лексему.

Следовательно, можно выделить набор атомарных операций, которые модифицируют текст программы неделимым образом. Определим трудоёмкость разработки каждого слова программы как количество атомарных операций, которые были сделаны для записи этого слова. Трудоёмкость написания текста программы — это суммарная трудоёмкость написания всех его слов, в том числе и тех, которые были удалены в процессе редактирования.

Оценим количество слов программы в зависимости от количества выполненных атомарных операций. Это и будет производительность труда.

## 2. Атомарные операции

Пусть  $L$  — формальный язык над некоторым алфавитом  $\Sigma$ . Множество всех слов над алфавитом  $\Sigma$  будем обозначать как  $\Sigma^*$ , а сами слова символами греческого алфавита  $\alpha, \beta, \gamma$ , длину слова  $\alpha$  — как  $|\alpha|$ . Напомним, что  $\Sigma^*$  является

полугруппой, то есть

$$\forall \alpha, \beta \in \Sigma^* : \alpha\beta \in \Sigma^*.$$

Пустое слово в  $\Sigma^*$  будем обозначать как  $\epsilon$ .

**Определение 1** (Удаление символа). Будем говорить, что слово  $\alpha'$  получено удалением символа  $a \in \Sigma$  из слова  $\alpha = \beta a \gamma$ , где  $\beta, \gamma \in \Sigma^*$ , причём одно из слов  $\beta$  или  $\gamma$  может быть пустым тогда и только тогда, когда  $\alpha'$  можно записать в виде  $\alpha' = \beta\gamma$ .

**Определение 2** (Добавление символа). Будем говорить, что слово  $\alpha'$  получено добавлением символа  $a \in \Sigma$  в слово  $\alpha = \beta\gamma$ , где  $\beta, \gamma \in \Sigma^*$ , причём одно из слов  $\beta$  или  $\gamma$  может быть пустым тогда и только тогда, когда  $\alpha'$  можно записать в виде  $\alpha' = \beta a \gamma$ .

**Пример 1.** Рассмотрим алфавит языка C++. Слово

```
void f()
{
    int
}
```

получено добавлением символа `int` из слова

```
void f()
{
}
```

И наоборот, второе слово может быть получено из первого удалением символа `int`.

Мы хотим ввести формальное определение процесса разработки над языком  $L$ . В рамках этого процесса могут появляться одинаковые слова из  $\Sigma^*$ . Чтобы их различать между собой, будем рассматривать не сами слова, а пары  $(\alpha, n)$ , где  $\alpha \in \Sigma^*$ ,  $n \in \mathbb{N}$ , и обозначать такие пары  $\bar{\alpha}$ , а само слово  $\alpha$  называть словом пары  $\bar{\alpha}$ , если будет необходимо.

Здесь и далее по тексту полагаем, что множество натуральных чисел  $\mathbb{N}$  содержит 0.

Пусть

$$S = \{\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n\},$$

— некоторое множество пар  $(\alpha_i, n_i)$ , где  $\alpha_i \in \Sigma^*$ ,  $n_i \in \mathbb{N}$ ,  $i \in \mathbb{N}$ ,  $i \in (1, 2, \dots, n)$ , при этом все  $n_i$  попарно различны между собой.

Определим на множестве пар  $S$  четыре типа операций:

1. Говорят, что множество пар  $S'$  получено добавлением в  $S$  пары  $(\alpha, k)$ , где  $\alpha \in \Sigma^*$ ,  $|\alpha| = 1$ ,  $k \in \mathbb{N}$ , если и только если

$$S' = S \cup \{(\alpha, k)\}, (\alpha, k) \notin S,$$

при этом  $\nexists (\beta, t) \in S : k = t$ .



2. Говорят, что множество пар  $S'$  получено из  $S$  удалением пары  $(\alpha, k)$ , где  $\alpha \in \Sigma^*$ ,  $k \in \mathbb{N}$ , если и только если

$$S' = S \setminus \{(\alpha, k)\}, (\alpha, k) \in S.$$

3. Говорят, что множество пар  $S'$  получено из  $S$  заменой пары  $(\alpha, k)$ , где  $\alpha \in \Sigma^*$ ,  $k \in \mathbb{N}$ , на пару  $(\alpha', k)$ , где слово  $\alpha'$  было получено из слова  $\alpha$  добавлением символа  $a \in \Sigma$  в смысле определения 3, если и только если

$$S' = S \setminus \{(\alpha, k)\} \cup \{(\alpha', k)\}, (\alpha, k) \in S.$$

4. Говорят, что множество пар  $S'$  получено из  $S$  заменой пары  $(\alpha, k)$ , где  $\alpha \in \Sigma^*$ ,  $k \in \mathbb{N}$ , на пару  $(\alpha', k)$ , где слово  $\alpha'$  было получено из слова  $\alpha$  удалением символа  $a \in \Sigma$  в смысле определения 2, если и только если

$$S' = S \setminus \{(\alpha, k)\} \cup \{(\alpha', k)\}, (\alpha, k) \in S.$$

**Замечание 1.**  $\forall \alpha \in \Sigma^*$  существует такая последовательность операций, результатом применения которых является пара  $(\alpha, n)$  для некоторого  $n \in \mathbb{N}$ .

*Доказательство.* Пусть  $\alpha \in \Sigma^*$ . Предположим, что  $\alpha = a_1 a_2 \dots a_n$ , где  $n > 0$ . Тогда возьмём следующий набор операций. С помощью операции типа 1 добавим пару  $(a_1, n)$ . Затем для каждого символа  $a_i, i > 1$  с помощью операции типа 3 заменим пару  $(a_1 a_2 \dots a_{i-1}, n)$  на пару  $(a_1 a_2 \dots a_{i-1} a_i, n)$ . ■

**Замечание 2** (Корректность набора операций). Пусть  $S$  — произвольное множество пар. Тогда существует последовательность операций, результатом применения которых является данное множество пар.

*Доказательство.* Выполняется индукцией по количеству пар в множестве  $S$  и применением предыдущего замечания к каждой паре. ■

### 3. Процесс разработки

Пусть  $\hat{L}$  — подмножество слов языка программирования  $L$  над некоторым алфавитом  $\Sigma$ . Подмножество  $\hat{L}$  символизирует те ограничения, которые накладывают программисты на используемый язык в связи с выбранной методологией программирования, например, процедурной, объектно-ориентированной и т. д. или стандартами кодирования, принятыми в рамках процесса разработки, например, запрет на использование глобальных переменных.

**Определение 3.** Пусть

$$S = \{\bar{\alpha}_i | i \in (1, 2, \dots, n)\},$$

где  $n \in \mathbb{N}$ ,  $\alpha_i \in \Sigma^*$  — некоторое множество пар.  $S$  называется программой на языке  $L$  тогда и только тогда, когда  $\forall \bar{\alpha}_i : \alpha_i \in \hat{L}$ .

**Определение 4.** Последовательность множеств пар:

$$P_0 \xrightarrow{c_1} P_1 \xrightarrow{c_2} P_2 \xrightarrow{c_1} \dots,$$

где  $P_0 = \emptyset$ ,  $P_i$ , где  $i > 0$  — множество пар,  $P_i$  получено из  $P_{i-1}$ , с помощью операции  $c_i$  одного из перечисленных выше типов, при этом для всех операций типа 1 все добавленные пары имеют вид  $(\alpha, k)$ , где  $\alpha \in \Sigma^*$ ,  $k$  — номер шага, на котором операция была выполнена, называется процессом разработки  $Pr$ .

**Определение 5.** Говорят, что программа  $P$  получена с помощью процесса разработки  $Pr$ , если  $\exists i > 0 : P_i = P$ .

Все дальнейшие рассуждения будем проводить в предположении, что имеется некоторый процесс разработки  $Pr$ :

$$P_0 \xrightarrow{c_1} P_1 \xrightarrow{c_2} P_2 \xrightarrow{c_1} \dots$$

**Замечание 3.** В силу определения процесса разработки для любого  $k \in \mathbb{N}$  либо не существует пары  $(\alpha, k)$ , которая была создана в процессе разработки, либо существует и при том только одна.

Определим для удобства последовательность множеств удалённых пар как:

1.  $D_0 = \emptyset$ ,
2.  $D_i = D_{i-1} \cup \{\bar{\alpha}\}$ , если  $\bar{\alpha}$  было удалено на шаге  $i$  из множества  $P_{i-1}$  с помощью операции типа 2, и  $D_i = D_{i-1}$  в противном случае.

**Предложение 1.**  $\forall n \in \mathbb{N} : P_n \cap D_n = \emptyset$ .

Доказательство проводится по индукции в силу построения множества пар  $P_n$ .

#### 4. Скорость процесса разработки

Далее определим на множестве  $\Sigma^* \times \mathbb{N}$  — декартовом произведении множеств  $\Sigma^*$  и  $\mathbb{N}$  — последовательность функций

$$F_i : (\Sigma^* \times \mathbb{N}) \rightarrow \mathbb{N}, i \in 1, 2, \dots, n :$$

1.  $F_0(\Sigma^* \times \mathbb{N}) = 0$ .
2. Предположим, что все функции  $F_j$ ,  $j \leq i$  определены. Пусть  $\bar{\alpha} \in \Sigma^* \times \mathbb{N}$  — некоторая пара. Тогда
  - $F_i(\bar{\alpha}) = 1$ , если пара  $\bar{\alpha} = (\alpha, i)$  была добавлена в  $P_i$  с помощью операции типа 1 на шаге  $i$ .
  - $F_i(\bar{\alpha}) = F_{i-1}(\bar{\alpha}') + 1$ , если пара  $\bar{\alpha} = (\alpha, k)$  была получена из  $\bar{\alpha}' = (\alpha', k)$  с помощью операции типа 3 или типа 4 на шаге  $i$ .

- $F_i(\bar{\alpha}) = F_{i-1}(\bar{\alpha}) + 1$ , если пара  $\bar{\alpha} = (\alpha, k)$  была удалена на шаге  $i$  из набора  $P_{i-1}$  с помощью операции типа 2.
- $F_i(\bar{\alpha}) = F_{i-1}(\bar{\alpha})$ , в противном случае.

**Предложение 2.**

$$\forall \bar{\alpha} \in \Sigma^* \times \mathbb{N} \forall k, m \in \mathbb{N} : k > m \Rightarrow F_k(\bar{\alpha}) \geq F_m(\bar{\alpha}).$$

Следует из построения набора функций  $F_i : \Sigma^* \times \mathbb{N} \rightarrow \mathbb{N}$ ,  $i \in \mathbb{N}$ .

**Предложение 3.** Пусть  $\bar{\alpha} \in \Sigma^* \times \mathbb{N}$ . Тогда последовательность  $\{F_i(\bar{\alpha}), i \in \mathbb{N}\}$  либо сходится к некоторому  $C \in \mathbb{N}$ , либо  $\forall C \in \mathbb{N} \exists k > 0 : F_k(\bar{\alpha}) > C$ .

Следует из определения функций  $F_i$ ,  $i \in \mathbb{N}$  и предложения 2.

**Определение 6.** Пусть  $\bar{\alpha} \in \Sigma^* \times \mathbb{N}$ . Тогда асимптотической трудоёмкостью написания слова  $\alpha$  в процессе разработки  $Pr$  называется  $\lim_{i \rightarrow \infty} F_i(\bar{\alpha})$ .

**Определение 7.** Определим  $F_{Pr} : \Sigma^* \times \mathbb{N} \rightarrow \mathbb{N}$  — асимптотическую трудоёмкость процесса разработки  $Pr$  как  $F_{Pr}(\bar{\alpha}) = \lim_{i \rightarrow \infty} F_i(\bar{\alpha})$ .

**Определение 8.** Пусть  $F : \mathbb{N} \rightarrow \mathbb{R}$  — некоторая функция из множества  $\mathbb{N}$  в  $\mathbb{R}$ . Говорят, что процесс разработки  $Pr$  обладает скоростью  $F$  тогда и только тогда, когда

$$\lim_{n \rightarrow \infty} \frac{|P_n|}{F(n)} = C,$$

где  $C > 0$ .

## 5. Достаточное условие линейной скорости процесса разработки

Пусть задан некоторый процесс разработки  $Pr$

$$P_0 \xrightarrow{c_1} P_1 \xrightarrow{c_2} P_2 \xrightarrow{c_1} \dots$$

В терминах предыдущего параграфа

**Предложение 4.**

$$\forall n \in \mathbb{N} |P_n| + |D_n| \leq n.$$

Доказательство утверждения выполняется индукцией по номеру  $n$ .

**Предложение 5.**

$$\forall n \in \mathbb{N} \sum_{\bar{\alpha} \in P_n \cup D_n} F_n(\bar{\alpha}) = n.$$

Доказательство утверждения выполняется индукцией по номеру  $n$ .

**Предложение 6.** Предположим, что  $\exists C > 0, C \in \mathbb{N} \forall n \in \mathbb{N} \forall \bar{\alpha} \in P_n \cup D_n : F_n(\bar{\alpha}) < C$ . Тогда

$$\frac{n}{C} \leq |P_n \cup D_n| \leq n.$$

*Доказательство.* Ограничение сверху следует из предложений 1 и 4. Ограничение снизу прямо следует из условия и предложения 5. ■

**Предложение 7.** Предположим, что  $\exists C > 0, C \in \mathbb{N} \forall n \in \mathbb{N} \forall \bar{\alpha} \in P_n \cup D_n : F_n(\bar{\alpha}) < C$ . Тогда

$$|P_n \cup D_n| \sim O(n).$$

*Доказательство.* Следует из предложения 6 и свойств предела числовой последовательности. ■

**Теорема 1** (Достаточное условие линейной скорости процесса разработки). В условиях предложения 7 предположим, что  $\lim_{n \rightarrow \infty} \frac{|D_n|}{|P_n|} = C$ , где  $C \geq 0$ . Тогда

$$|P_n| \sim O(n).$$

*Доказательство.* Следует из предложений 1, 7 и свойств предела числовой последовательности. ■

Рассмотрим теперь случай, когда асимптотическая трудоёмкость разработки каждого слова ограничена некоторой константой, но при этом не существует такой константы, которая ограничивает сразу асимптотическую трудоёмкость разработки всех слов. Представим данное условие в виде двух высказываний:

$$\forall n \in \mathbb{N} \forall \bar{\alpha} \in P_n \cup D_n \exists C > 0 : F_n(\bar{\alpha}) < C. \quad (1)$$

$$\forall C > 0 \exists k \in \mathbb{N} \exists \bar{\beta} \in P_k \cup D_k : F_k(\bar{\beta}) > C. \quad (2)$$

Пусть  $C > 0$  — некоторая константа. Построим множество  $S$  пар  $\bar{\alpha}$  следующим образом:

1. В силу предположения (2)  $\exists k \in \mathbb{N} \exists \bar{\beta} \in P_k \cup D_k : F_k(\bar{\beta}) > C$ . Тогда определим  $S_0 = \{\bar{\beta}\}$ .
2. Предположим, что для всех  $j < i$  множества  $S_j$  уже построены. Среди всех  $\bar{\beta} : \exists k \in \mathbb{N} \exists \bar{\beta} \in P_k \cup D_k : F_k(\bar{\beta}) > C + i$  выберем, что  $\forall j < i : \bar{\beta} \notin S_j$ . Если такое  $\bar{\beta}$  существует (доказательство существования см. в предложении 8 ниже по тексту), то определим  $S_i = S_{i-1} \cup \{\bar{\beta}\}$ .

**Замечание 4.**

$$\forall i \in \mathbb{N} : |S_i| = i.$$

**Замечание 5.**  $\forall i > 0 : S_{i-1} \subset S_i$ .

**Предложение 8.** В условиях пункта 2 построения множества  $S$   $\bar{\beta}$  существует.

*Доказательство.* От противного. Предположим, что  $\bar{\beta}$  не существует, тогда в силу (2), замечания 5 и сделанного предположения

$$\forall \bar{\gamma} \exists k \in \mathbb{N} : (\bar{\gamma} \in P_k \cup D_k) \wedge (F_k(\bar{\gamma}) > C + i) \rightarrow \bar{\gamma} \in S_{i-1},$$

где  $\rightarrow$  — это импликация. Поскольку множество  $S_{i-1}$  — конечное, и для каждой пары  $\bar{\alpha}$  из  $S_{i-1}$  в силу (1) существует константа  $C_\alpha$ , что  $F_n(\bar{\alpha}) < C_\alpha$ . Тогда мы можем определить общую константу  $C_g = \max_{\bar{\alpha} \in S_{i-1}} C_\alpha$ , что  $\forall n \in \mathbb{N} \forall \bar{\alpha} \in P_n \cup D_n : F_n(\bar{\alpha}) < C_g$ . Получаем противоречие с (2), следовательно,  $\bar{\beta}$  существует. ■

Определим множество  $S$  как

$$S = \bigcup_{i \in \mathbb{N}} S_i.$$

**Предложение 9.** Множество  $S$  бесконечное.

*Доказательство.* Следует из построения множества  $S$  в силу замечания 5. ■

**Теорема 2.** Пусть (1) и (2) справедливы для процесса разработки. Тогда

$$\forall C > 0 \exists k \in \mathbb{N} : |P_k \cup D_k| > C.$$

*Доказательство.* Пусть  $C > 0$  — некоторая константа,  $S$  — множество, построенное, как описано выше. В силу предложения 9  $\exists k \in \mathbb{N} : |S_k| > C$ . В силу построения множества  $S : \exists m \in \mathbb{N} : S_k \subset P_m \cup D_m$ . Следовательно,  $|P_m \cup D_m| > C$ . ■

Условий (1) и (2) недостаточно, чтобы процесс разработки обладал линейной скоростью. Рассмотрим следующий пример.

Пусть  $C$  — некоторая константа. Рассмотрим процесс разработки  $Pr_{ex}$ , при котором последовательно был получен следующий набор слов:

$\bar{\alpha}_1 : F_{Pr}(\bar{\alpha}_1) = C, \bar{\alpha}_2 : F_{Pr}(\bar{\alpha}_2) = C+1, \bar{\alpha}_3 : F_{Pr}(\bar{\alpha}_3) = C+2, \dots$  Следовательно,

$$\sum_{i=1}^k F_k(\bar{\alpha}_i) = Ck + \frac{k(k-1)}{2}. \quad (3)$$

Тогда набор пар

$$(\alpha_1, 1), (\alpha_2, C+1), (\alpha_3, C+2), \dots, \left(\alpha_k, C + \frac{k(k-1)}{2}\right)$$

составляет множество пар  $P_n$  для некоторого шага  $n$  процесса  $Pr$ . Заметим, что  $|P_n| = k$ . В силу предложения 5 и формулы (3)

$$Ck + \frac{k(k-1)}{2} = n.$$

Нетрудно показать, что в таком случае имеет место

$$|P_n| \sim O(\sqrt{n}).$$

Другими словами, рассмотренный нами в примере процесс разработки  $Pr_{ex}$  имеет скорость  $\sqrt{n}$ .

**Предложение 10.** Процесс разработки  $Pr_{ex}$  существует.

*Доказательство.* Рассмотрим функцию

```
void f1()  
{  
}
```

Данная функция состоит из 7 лексем и согласно нашему определению трудоёмкости при условии, что мы только последовательно добавляли лексемы, её трудоёмкость составляет  $6+1$ .

Пусть  $k \in N, k > 1$ . Рассмотрим функцию

```
void f1...1()  
{  
    ; ; . . . ;  
}
```

Имя этой функции состоит из  $f$  и  $k$  единиц, а тело функции содержит  $k$  пустых операторов. То есть данная функция состоит из  $6 + k$  лексем и, следовательно, при условии, что мы последовательно добавили  $k$  лексем, трудоёмкость её разработки составит  $6 + k$ . Мы показали, как построить функцию для любого заданного  $k$ . ■

## 6. Полученные результаты и практика

Достаточное условие линейной скорости процесса разработки (Теорема 1) подтверждается практиками разработки программного обеспечения:

- Мартин Фаулер в [4] с. 54–56 даёт рекомендацию: «классы, процедуры, функции должны быть небольшими».
- The Open-Closed Principle [5]: «Программные сущности (классы, модули, функции и т.п.) должны быть закрыты от изменений...»

Важность ограничений на модификацию программного кода легко проиллюстрировать на следующем примере. Рассмотрим рисунок 2. Прямоугольники обозначают процедуры, а стрелки — какие-либо зависимости одной процедуры от другой. Например, если одна процедура содержит в своём теле вызов второй, то первая процедура зависит от второй. Основание стрелки — это процедура, которая зависит от второй, на которую показывает стрелка. Предположим, что нам необходимо внести изменения в код процедуры, обозначенной цифрой 1. Кроме тестирования самой процедуры 1, нам придётся, как минимум, протестировать, а, возможно и вносить правки в код процедур 2 и 3. В случае правок в процедуре 2, придётся проверять, и, возможно изменять процедуру 4, а в случае правок в процедуре 3 — выполнить те же действия для процедур 5 и 6. И так далее.

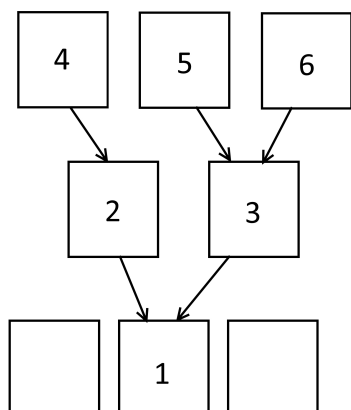


Рис. 2. Расходящиеся модификации

Этот пример показывает, что при внесении изменений в какой-либо код могут понадобиться правки в коде, который использует модифицируемый. При этом объём кода, работоспособность которого может быть нарушена, не зависит и никак не может быть ограничен разработчиком кода, для которого потребовалась модификация.

Существует немало примеров, когда небольшие правки затрагивали большое число программных проектов по всему миру:

- Проблема 2000 года. Проблема была связана с форматом хранения даты, который был рассчитан на хранение диапазона дат с 1970 по 1999 годы. Никто не мог сказать, как наступление 1 января 2000 года отразится на работоспособности программного обеспечения. Серьёзных последствий удалось избежать лишь потому, что эта проблема была поднята за несколько лет до наступления 2000 года: и производители оборудования, а затем и программисты с учётом темпов смены технологий заблаговременно внесли изменения в программный код.
- Февраль 2016 года [7]. Разработчик функции, которая отрезает пробелы слева, удалил соответствующий npm пакет из репозитория javascript пакетов. В результате существенное число популярных javascript проектов перестало собираться. Многие разработчики даже не догадывались, что их проекты зависят от этой функции, так как зависимость была у сторонних проектов, включённых в проект.
- Страуструп, создатель языка C++, в [6] признается, что некоторые полезные и изящные конструкции не вошли в состав стандарта языка C++, так как они могли нарушить обратную совместимость.

## Заключение

В настоящей статье было определено достаточное условие линейной скорости процесса разработки программного обеспечения, которое не зависит от языка и используемой парадигмы программирования, и которое, в основном, сводится к требованию, чтобы на разработку каждой программной сущности: класса, процедуры, функции и т. д. – не должно уходить более, чем заранее фиксированное число операций.

Кроме этого, показано, что требование, чтобы каждая программная сущность лишь не модифицировалась, то есть имела конечную трудоёмкость разработки, недостаточно, чтобы процесс разработки обладал линейной скоростью.

## ЛИТЕРАТУРА

1. Колесов А. Мифический человеко-месяц: двадцать пять лет спустя // PCWeek. 2000. № 7(229). URL: <https://www.pcweek.ru/themes/detail.php?ID=53642>.
2. Чапел Х., Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы. СПб. : Символ-Плюс, 2010.
3. Nanus B., Farr L. Some cost contributors to large-scale programs // AFIPS Proc. SJCC. Spring 1964. Vol. 25. P. 239–248.
4. Фаулер М. Рефакторинг: улучшение существующего кода. СПб. : Символ-Плюс, 2003.
5. Martin R. The Open-Closed Principle // Object Mentor. 1996. URL: <http://www.objectmentor.com/resources/articles/ocp.pdf>.
6. Страуструп Б. Дизайн и эволюция языка C++. СПб. : ДМК Пресс, 2006.
7. Williams C. How one developer just broke Node, Babel and thousands of projects in 11 lines of JavaScript. URL: [https://www.theregister.co.uk/2016/03/23/npm\\_left\\_pad\\_chaos/](https://www.theregister.co.uk/2016/03/23/npm_left_pad_chaos/).

## ABOUT THE FORMALIZATION OF THE SOFTWARE DEVELOPMENT PROCESS

Е.А. Tyumentcev

CEO, e-mail: [etyumentcev@hwdtech.ru](mailto:etyumentcev@hwdtech.ru)

LLC Hello World! Technologies

**Abstract.** In the book “Mythical man-month, or How software systems are created,” Brooks refers to several studies on the basis of which it can be concluded that the productivity of the work of programmers, measured in the number of lines of code per time, decreases as the size of the program project grows. In this paper, we describe the formalization of the software development process, as the process of editing the source code of the program, with the help of which the laboriousness function is defined, and a sufficient condition for the constant productivity of the programmers, independent of the size of the project, is derived.

**Keywords:** formalization, development process, software, productivity, Brooks, mythical man-month.

*Дата поступления в редакцию: 25.04.2017*



## **ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА МЕТОДОМ К-СРЕДНИХ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТОВ НАУЧНОЙ НАПРАВЛЕННОСТИ**

**Ю.А. Осипова**

студент, e-mail: fray09@inbox.ru

**Д.Н. Лавров**

к.т.н., доцент, e-mail: dmitry.lavrov72@gmail.com

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** В статье рассматриваются вопросы построения программного обеспечения для выработки рекомендаций по назначению УДК. В основу разработки положен метод к-средних.

**Ключевые слова:** метод к-средних, лемматизация, TF-IDF.

### **Введение**

Существует проблема определения принадлежности статьи к какой-либо из тематик или рубрик журнала и присвоения наиболее подходящего номера универсальной десятичной классификации (УДК). Очень часто авторы научных статей назначают УДК своей статье не на основе существующих справочников, а по примеру статьи схожей тематики. Тем не менее, тематика может немного меняться. Такой подход к назначению УДК после цепочки переприсвоений приводит к тому, что тема последней статьи может совсем не соответствовать назначенному УДК первой статьи. УДК у таких статей будет одинаковым. Решение простое — редакторы журнала должны проверять УДК, назначенные авторами статей.

Для автоматизации этого процесса необходимо разработать программное обеспечение классификации научных текстов для выработки рекомендаций по назначению первичного УДК научной статье.

Для достижения этой цели в работе решаются следующие задачи:

1. Провести сравнительный обзор существующих методов лемматизации текстов.
2. Расширить функционал программного обеспечения путём внедрения выбранных методов лемматизации.
3. Провести эксперименты для подтверждения работоспособности разработанного ПО.

## 1. Обзор существующих решений

### 1.1. Основные определения

Рассмотрим основные термины и определения, необходимые для дальнейшего изложения материала.

*Лемма* — основная или словарная форма слова. Например, для русского языка это именительный падеж единственного числа для существительного, инфинитив для глагола и т. д.

*Лемматизация* — это процесс, который использует морфологический анализ для приведения слова к лемме (нормальной форме). Также включает в себя определение лексической категории и применение различных правил нормализации для каждой части речи.

*Лемматизатор* — программный продукт для обработки естественного языка, который, используя правила лемматизации, возвращает начальную форму слова. Отдельную сложность составляет сама структура нашего языка. Русский язык преимущественно флективный, что означает и высокую его гибкость и аналитическую сложность. Особенность языков данного типа в том, что словоизменение происходит с помощью флексий — формантов, несущих в себе множество значений. Например, род, число и падеж для окончаний. Такая многовариантность обуславливает трудность определения леммы в большом количестве словоформ.

Определим критерии, на которых необходимо сосредоточить внимание при сравнении рассматриваемых методов:

1. Язык. Возможно ли использование данного решения для лемматизации текстов на русском языке.
2. Метод. То, каким образом метод приходит к результату.
3. Формат результата. В зависимости от метода меняются и возвращаемые данные: в одном случае — это основа слова, в другом — полная морфологическая информация.
4. Универсальность — может работать со всеми кодировками, типами данных и т. д.
5. Отдельное внимание стоит уделить тому, какую задачу решает данный метод.

Проанализируем существующие решения по заданным параметрам. Это позволит выбрать наиболее подходящую стратегию для решения поставленной задачи.

### 1.2. Стемминг

*Стемминг* — эвристический процесс, который находит основу для заданного слова. Основа не обязательно должна совпадать с морфологическим корнем слова.

Стемминг используется поисковыми системами для расширения запросов — *query expansion (QE)*, а также является частью процессов лемматизации текстов на естественном языке. QE необходим для переформулировки запросов

для улучшения производительности в системах поиска информации, особенно в контексте понимания запросов.

Существует множество вариаций алгоритма стемминга. Один из наиболее подходящих для решения поставленной задачи — алгоритм усечения окончаний.

Данный алгоритм содержит список правил, который используется для нахождения основы слова с учётом его формы. Большинство из них выглядит следующим образом:

- если конец слова = «ый», обрезать «ый»;
- если конец слова = «ть», обрезать «ть»;
- если конец слова = «ому», обрезать «ому».

За счёт отсутствия справочных таблиц и полного перебора обеспечивается эффективность алгоритма по отношению к другим представителям семейства. Также за счёт настройки правил он работает с флективными языками. Однако это не означает, что полученный результат будет означать существование такого слова в лексиконе языка.

### 1.3. Stanford CoreNLP

Stanford CoreNLP — набор инструментов для анализа естественного языка. Он используется для получения форм слова, их частей речи; также для нормализации даты, времени и числовых величин; для разметки структуры предложения (указывает, какие существительные относятся к одним и тем же объектам).

Основные преимущества Stanford CoreNLP:

- быстрый и надёжный анализ текста;
- общая аналитика самого высокого качества;
- поддержка ряда основных естественных языков;
- доступные интерфейсы для большинства современных языков программирования;
- интегрированный инструментарий с хорошим набором инструментов грамматического анализа.

Stanford CoreNLP легко применяет множество инструментов лингвистического анализа к фрагменту текста. Код библиотеки представлен на языке Java и разрешён для использования в некоммерческих проектах под лицензией GPL.

С помощью настроек можно изменить, какие опции и инструменты должны быть включены, а какие отключены. CoreNLP объединяет многие инструменты: POS-тэгер, распознаватель названной сущности (NER), синтаксический анализатор (парсер), систему разрешения связей, анализатор тональности.

### 1.4. AOT

Проект AOT — набор инструментов, используемых для автоматической обработки текстов на естественном языке. В основном применяется для русского языка.

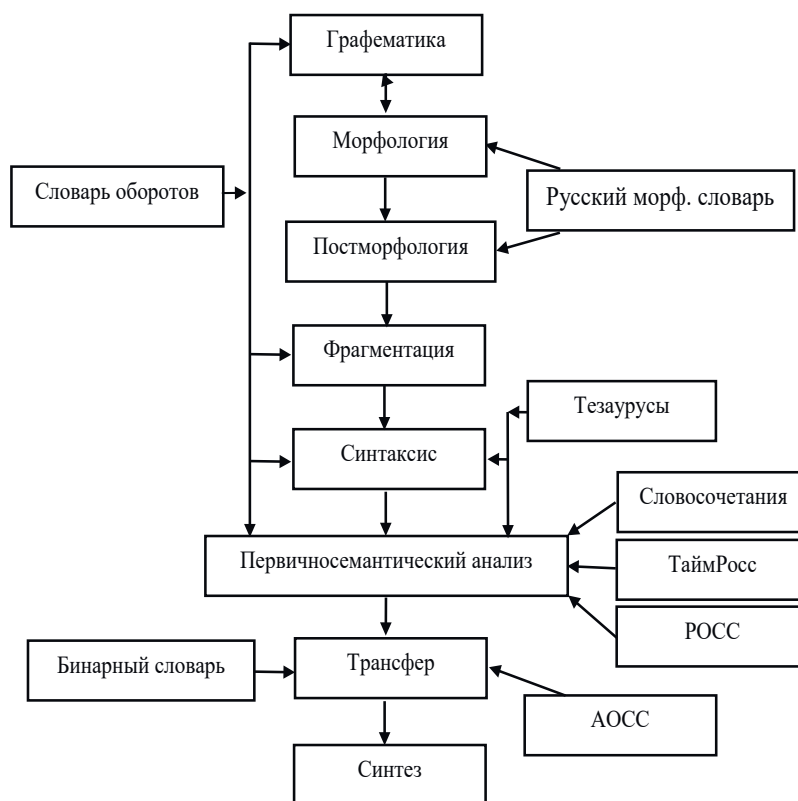


Рис. 1. Компоненты проекта АОТ

Суть данного механизма — в грамотной декомпиляции языковых механизмов, которая позволяет максимально приблизить человеческий язык к языку, понятному компьютеру.

Набор состоит из множества компонентов, которые один за другим обрабатывают исходный текст (см. рис. 1). Вход одного лингвистического процесса является выходом другого. Выделяются такие процессоры как графематический, морфологический, синтаксический и семантический [1]. Основан на грамматическом словаре А.А. Зализняка (1987). На данный момент содержит 161 тыс. лемм.

Для каждого слова входного текста формирует набор морфологических интерпретаций следующего типа:

- лемма;
- морфологическая часть речи;
- набор общих граммем (относящихся ко всевозможным словоформам парадигмы слова);
- многообразие наборов граммем.

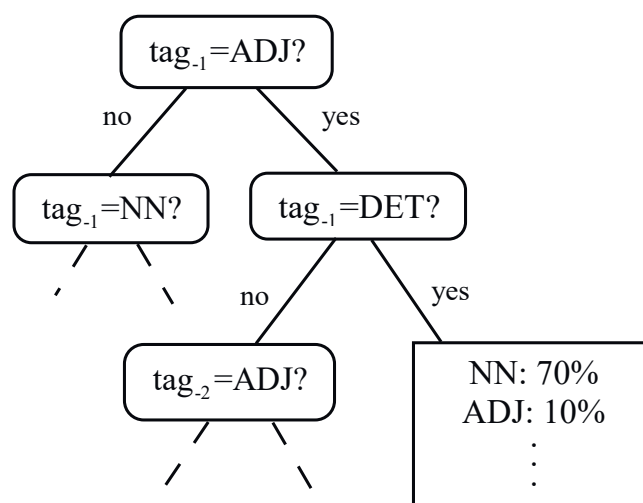


Рис. 2. Пример дерева решений Tree Tagger

### 1.5. Tree Tagger

Инструмент разработан Хелмутом Шмидом в университете Штутгарта. Это языконезависимый метод морфологической разметки текстов на естественном языке. Он основан на деревьях принятия решений и применяется в задачах обработки текстов на множестве языков. Исходный код данного метода закрыт. Распространяется в виде архива с исполняемыми файлами и набором языковых компонент.

Суть алгоритма в том, что оценка вероятности того, какой частью речи является слово, определяется бинарным деревом принятия решений. Вероятность определяется следующим образом: путь внутри дерева считается до тех пор, пока лист не будет достигнут [2]. На ветвях дерева записаны условия, от которых зависит целевая функция, а на листьях — значения этой функции.

Рассмотрим пример на рис. 2, который демонстрирует образец дерева решений. Вероятность данной триграммы определяется следующим образом: продолжать процесс, пока операция не достигнет листа дерева. Например, если мы ищем, с какой вероятностью существительному предшествует определяющее слово и прилагательное, сперва мы должны ответить на вопрос в корневом узле. Если тег предыдущего слова — прилагательное, то выбираем положительную ветвь. Следующий ответ тоже положительный и, таким образом, алгоритм достигает конечного узла. Теперь остаётся лишь узнать вероятность тега  $n$  в таблице, прикреплённой к коду.

### 1.6. Сравнение рассмотренных решений

Были рассмотрены основные алгоритмы лемматизации текстов на естественном языке. В таблице 1 приведены общие сведения, необходимые для сравнения алгоритмов по параметрам, приведённым в начале обзора:

Таблица 1. Сравнение существующих алгоритмов лемматизации

| Алгоритм                     | Русский язык | Метод                   | Формат результата  | Универсальность         | Задача                     |
|------------------------------|--------------|-------------------------|--------------------|-------------------------|----------------------------|
| Алгоритм усеечения окончаний | Есть         | Стемминг                | Основа слова       | —                       | Поисковые системы          |
| Stanford Core NLP            | Нет          | Словарный               | Лемма              | UTF-8                   | Анализ текстов             |
| АОТ                          | Есть         | Словарный               | Лемма + часть речи | UTF-8, только кириллица | Машинный перевод           |
| Tree Tagger                  | Есть         | Дерево принятия решений | Лемма + часть речи | —                       | Анализ и обработка текстов |

- алгоритм — название алгоритма лемматизации;
- русский язык — возможность использования решения для текста на русском языке;
- метод — какой метод используется данным решением;
- формат — вид полученного результата в ходе работы системы;
- универсальность — особенности входных данных для текущего решения;
- задача — изначальное назначение данной системы.

Из таблицы видно, что нет явного лидера. Кроме того, неясна скорость работы этих методов. Поэтому было принято решение об использовании нескольких методов лемматизации для достижения поставленной цели.

## 2. Реализация

Задача разрабатываемого программного обеспечения — помощь в присвоении номера универсальной десятичной классификации. Результат работы программы должен быть представлен в виде списка предпочитаемых УДК со степенью соответствия УДК тексту статьи

Основные этапы работы разработанного ПО представлены на рис. 3.

Построение системы логически разбивается на две задачи: подготовка входных данных, необходимых для кластеризации, и собственно кластеризация.

### 2.1. Подготовка данных

#### 2.1.1. Получение текстов

На этом этапе необходимо получить данные из заранее собранной базы текстов sqllite в формате: идентификационный номер, название, содержание, номер УДК.

База сформирована из 150 текстов, опубликованных в журнале «Математические структуры и моделирование» с 2010 года, у которых определён номер УДК редакторами.

Затем каждый документ передаётся в список и хранится там для дальнейшей обработки.

С помощью графического интерфейса (рис. 4) вводится текст, УДК которого необходимо узнать. Также определяются параметры, необходимые для дальнейшей работы программы.

### 2.1.2. Лемматизация

На данном этапе нормализуется каждое слово в текущем списке текстов. Для этого выбранному пользователем лемматизатору на вход подаётся содержание каждого текста из листа. Во время этого процесса отбрасываются стоп-слова, не имеющие смысловой нагрузки. Например, предлоги, местоимения, союзы, междометия и т. д.

Затем содержание каждого документа токенизируется и отдельные слова обрабатываются непосредственно алгоритмом, с учётом особенностей каждого лемматизатора. После чего из полученного списка слов формируется корпус слов всех текстов.

$$corp = \begin{bmatrix} w_1, \dots, w_i \\ l_1, \dots, l_i \end{bmatrix},$$

где  $w$  — слова, содержащиеся в корпусе, а  $l$  — лист документов, в которых оно содержится.

Для каждого элемента списка корпуса подсчитывается количество текстов, в которых он встречается.

По завершении этого процесса отсекаются слова с наименьшим употреблением для ускорения работы алгоритма и уменьшения вектора корпуса, используемого для каждого документа.

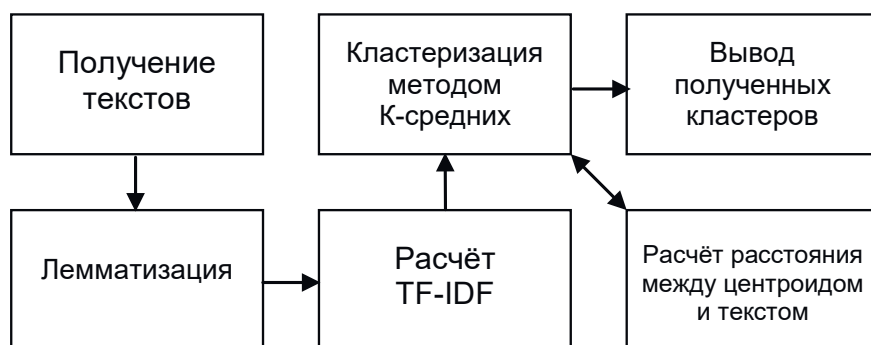


Рис. 3. Этапы работы разработанного ПО

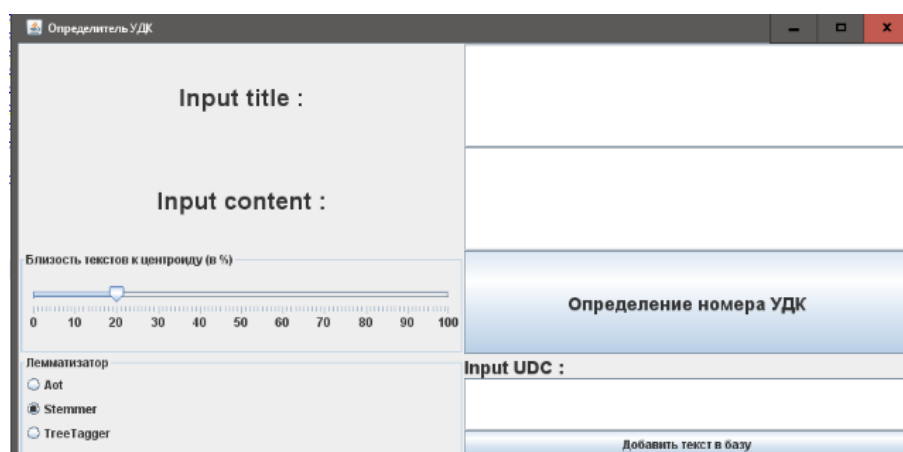


Рис. 4. Графический интерфейс

### 2.1.3. Расчет TF-IDF

С помощью кодировщика, использующего TF-IDF, содержание каждого документа кодируется в вектор. TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency) — мера статистического анализа важности слова в контексте документа. Используется при расчёте близости текстов друг к другу. Состоит из двух частей TF и IDF.

TF (от англ. — частота слова) — численное значение вхождения заданного слова в текущем документе. Рассчитывается по формуле:

$$TF = \frac{n_i}{\sum n_k},$$

где  $n_i$  — количество вхождений данного слова,  $n_k$  — общее число слов в документе.

IDF (от англ. — обратная частота документа) — численное значение, показывающее с какой частотой данное слово встречается во всех исходных документах. Расчётная формула:

$$IDF = \log\left(\frac{D}{d_i}\right),$$

где  $D$  — общее количество документов, а  $d_i$  — документы, в которых встречается данное слово.

Конечное значение коэффициента TFIDF равно произведению выше представленных множителей

$$TFIDF = TF \cdot IDF.$$

Большой вес получают слова с высокой частотой в пределах данного документа и с низкой частотой в пределах всего набора документов.

Для расчёта TF используется вектор нормализованных слов. Каждое слово становится ключом в карте, а количество вхождений — значением.



За основу IDF берётся вектор корпуса слов, полученный ранее. Его длина задаёт размерность вектора TF-IDF для каждого документа. Это необходимое условие при расчёте расстояния между двумя векторами.

Затем определяется метрика, необходимая для расчёта расстояния. Так как выбранный метод  $k$ -средних, то и метрика — евклидово расстояние, которое рассчитывается по формуле:

$$d = \sqrt{\sum_i^n (x_i - x'_i)^2},$$

где  $x$  и  $x'$  — точки, между которыми рассчитывается расстояние.

На этом заканчивается подготовительный этап и начинает работу алгоритм кластеризации.

## 2.2. Кластеризация

После этого полученные данные передаются непосредственно алгоритму кластеризации. Именно он участвует в формировании групп сходных по смыслу текстов.

В качестве алгоритма кластеризации выбран метод  $k$ -средних.

Алгоритм  $k$ -средних — один из наиболее популярных и простых в реализации методов кластерного анализа. Это неиерархичный метод неконтролируемого обучения. Он позволяет разделить произвольный набор данных на заданное число кластеров так, что объекты внутри одного кластера были достаточно близки друг к другу, а объекты разных не пересекались. Другими словами, цель этого алгоритма — объединить в группы сходные данные по некоторым заданным критериям (см. рис. 4). Чаще всего при кластеризации используются меры расстояния. По умолчанию данный алгоритм использует квадрат Евклидова расстояния.

В 1950-х годах независимо друг от друга Гуго Штейнгауз и Стюарт Ллойд опубликовали свои работы [3, 4]. Однако особую популярность метод получил после работы Маккуина. Детальное описание и возможные изменения даны Джейном и Дабсом.

Алгоритм итеративен, то есть повторяется до тех пор, пока не будет выполнено условие выхода. Начнём с того, что на всём множестве доступных значений  $D$  выберем  $k$  центроидов (центров масс). Затем каждую точку из множества  $D$  привязываем к ближайшему центроиду. После того как все точки распределены, пересчитываем центры масс для каждого кластера. А дальше идёт повтор двух предыдущих действий до тех пор, пока центроиды не перестанут сдвигаться или действие алгоритма искусственно не прервётся.

К достоинствам данного алгоритма относятся следующие характеристики. Во-первых, результат работы алгоритма не зависит от порядка следования данных. Во-вторых, простота реализации.

Недостатки метода следующие. Первый — необходимость заранее задавать число кластеров. Но эту проблему достаточно просто решить. Одно из таких

возможных решений — повторить несколько раз алгоритм при предполагаемых значениях. И затем выбрать наилучшую кластеризацию. Также несколько решений этой проблемы представлены в книге [5].

В текущей реализации для решения проблемы с необходимостью предварительной передачи количества кластеров число кластеров  $k$  устанавливается в наименьшем значении, которое обеспечивает расстояние от внешнего кластера до внутреннего. Данное расстояние рассчитывается как отношение среднего расстояния между кластерами к среднему расстоянию между документами в кластере. Таким образом, нам необходимо задать только кластеризационный порог, относительно которого и будет строиться необходимое число кластеров. После чего на каждой итерации цикла будет рассчитано распределение документов для текущего числа кластеров до тех пор, пока расстояние между текущим количеством кластеров не станет ниже кластеризационного порога.

Также для более тонкой настройки пользователю даётся возможность выбрать расстояние, показывающее насколько близко к центру кластера должны располагаться вектора документов, чтобы считаться частью кластера.

Ниже приведены этапы работы алгоритма.

- Центроидом первого кластера становится TF-IDF вектор того документа, УДК которого необходимо определить. Заполняем остальные  $k$  кластеров достаточно удалёнными документами из общего набора. Их нормализованные вектора слов — начальные центроиды.
- Распределяем оставшиеся нераспределёнными документы в прилагаемом списке документов по ближайшим кластерам, предоставляемым набором кластеров.
- Обновляем центроиды кластеров в предоставляемом наборе кластеров.
- Повторяем шаги 2 и 3 до тех пор, пока число итераций не будет равно заданному выше кластеризационному порогу.

В результате работы данного алгоритма получаем кластеры со сходными по содержанию текстами. После чего пользователю предоставляются полученные данные в наглядной форме (см. рис. 5).

### 3. Эксперимент

#### 3.1. Сравнение алгоритмов лемматизации

Для эксперимента по определению оптимального лемматизатора была собрана база из 150 текстов, опубликованных в журнале «Математические структуры и моделирование» с 2010 года, у которых определён номер УДК редакторами журнала. Затем был выбран текст того же журнала выпуском ранее 2010 года. Общее число слов равно 306215. Число слов, обработанных лемматизатором равно 198844. Число запусков равно 100.

Большая разница между исходным числом слов и слов, подающихся лемматизатору, обусловлена сложностью обработки формул. Большинство из них сформировано с помощью функций в  $\text{\LaTeX}$ , поэтому они некорректно отображаются непосредственно в самой базе данных и целевом тексте. При предва-

| Name    | Count | Description                |
|---------|-------|----------------------------|
| 004     | 40,24 | Информационные техно...    |
| 004.0   | 6,10  |                            |
| 004.05  | 6,10  | Качество систем и прогр... |
| 004.051 | 1,22  | Эффективность              |
| 004.056 | 4,88  | Безопасность, защищён...   |
| 004.3   | 3,66  | Аппаратные средства. Т...  |
| 004.38  | 1,22  | Виды компьютеров           |
| 004.4   | 4,88  | Программные средства       |
| 004.5   | 1,22  | Человеко-машинное вза...   |
| 004.6   | 1,22  | Данные                     |
| 004.7   | 2,44  | Связь компьютеров. Сет...  |
| 004.8   | 4,88  | Искусственный интеллект    |
| 004.9   | 13,41 | Прикладные информац...     |
| 051     | 1,22  | Математика                 |
| 051.7   | 1,22  |                            |
| 051.74  | 1,22  |                            |
| 316     | 2,44  | Социология                 |
| 316.4   | 2,44  | Социальные процессы. ...   |
| 372     | 3,66  | Содержание и форма д...    |
| 378     | 1,22  | Высшее образование. В...   |
| 504     | 1,22  | Науки об окружающей с...   |
| 510     | 2,44  | Фундаментальные и об...    |
| 512     | 2,44  | Алгебра                    |

Рис. 5. Данные, предоставляемые пользователю

Таблица 2. Результаты работы на тестовых текстах

| Название алгоритма | Среднее время выполнения (в сек.) | Скорость (слов/сек.) | Выходное число слов корпуса (-5% низко употребляемых) |
|--------------------|-----------------------------------|----------------------|-------------------------------------------------------|
| Стеммер            | 2,48                              | 80179                | 2020                                                  |
| Аот                | 55,42                             | 3588                 | 2006                                                  |
| Tree Tagger        | 759,59                            | 262                  | 2365                                                  |

рительной обработке они отсекаются вместе со стоп-словами для корректного выполнения программы.

В таблице 2 собраны результаты на тестовом наборе текстов. Стеммер подтвердил свою скорость. Рекомендуется использовать в качестве лемматизатора Стеммер и АОТ, т. к. временные затраты на их использование существенно меньше.

Для повышения точности необходима более обширная база текстов, с несколькими представителями каждого номера УДК, используемого в журнале. Также есть вероятность ошибки в определении номера в уже обработанных текстах (см. рис. 4), что повышает вероятность погрешности.

### 3.2. Проверка работоспособности разработанного ПО

Для эксперимента по определению эффективности работы программы взята уже собранная база из предыдущего эксперимента и статьи на русском языке из ещё не опубликованного номера журнала «Математические структуры и моделирование».

Из 13 полученных текстов в 11 результат первого уровня УДК совпал с присвоенным редактором, что составляет 84,6% верно проведённых вычислений. Для дальнейшей оценки используем 11 оставшихся статей, в 6 верно определён номер универсальной десятичной классификации второго уровня, что составляет 55% корректной оценки.

Причиной таких результатов являются:

1. Использование формул и специальных символов, не воспринимаемых текстовым редактором.
2. Наличие типичных для данного журнала УДК, например, 510 и 514. Для рассматриваемого журнала характерны такие номера как 004, 517 и 519, а также их подуровни. Для решения этой проблемы необходимо расширить текстовую базу. Существуют номера журнала «Математические структуры и моделирование», вышедшие ранее 2010 года, однако, опубликованные в них статьи не проходили проверку редактором.
3. Также в одном из текстов изначально не был определён УДК ниже второго уровня, что не позволяет говорить о точности работы программного обеспечения (ПО) в отношении этой статьи.

В ходе экспериментального использования ПО была подтверждена эффективность данной разработки, так же выявлены проблемы текущего этапа разработки и обозначены пути их решения.

## Заключение

Программная реализация метода  $k$ -средних подтвердила его эффективность, однако, подтвердилась и основная проблема при практическом использовании метода — сложность в выборе количества итераций. Слишком большое число повторений может привести к тому, что сложность алгоритма увеличится до  $O(k^2)$ . Она может быть решена практическим перебором необходимого числа итераций, а также предварительным анализом данных.

В ходе работы были достигнуты следующие результаты.

1. Проведён сравнительный обзор существующих методов лемматизации текстов.
2. Расширен функционал программного обеспечения путём внедрения выбранных методов лемматизации.
3. Проведены эксперименты, подтверждающие работоспособность разработанного ПО.

В качестве дальнейшего развития можно рассмотреть следующие направления.

Таблица 3. сводные данные по эффективности работы

| N  | Заголовок                                                                                                                          | Присвоенный номер УДК | Рекомендованный номер УДК |
|----|------------------------------------------------------------------------------------------------------------------------------------|-----------------------|---------------------------|
| 1  | Моделирование и оптимизация процесса распределения человеческих ресурсов и аудиторного фонда вуза на основе анализа учебных планов | 004.912+021           | 004.0                     |
| 2  | Применение метода построения ассоциативных правил к анализу деятельности общественных организаций                                  | 004.93                | 004.9                     |
| 3  | О редукции модальностей в деонтических исчислениях                                                                                 | 510.643               | 519.2                     |
| 4  | Разработка системы мандатного управления доступом для операционных систем семейства Windows                                        | 004.42                | 004.4                     |
| 5  | Вопросы существования устойчивого цикла в одной модели молекулярного репрессилатора                                                | 514.745.82            | 517.9                     |
| 6  | Идентификация параметрической модели изменения численности зачисленных абитуриентов по малой выборке                               | 004.000               | 004.9                     |
| 7  | Равновесная динамика лесных экосистем с учётом взаимосвязи «растительность-почва»                                                  | 004.9:631.4 + 519.6   | 519.2                     |
| 8  | Аналитическое и численное представление звуковых характеристик обобщённой центрированной волны                                     | 519.6                 | 519.7                     |
| 9  | Введение зависимости в схему испытаний Бернулли                                                                                    | 519.2                 | 519.2                     |
| 10 | Латентный анализ на базе метода штрафных функций для многомерных бинарных показателей                                              | 519.237.07            | 519.1                     |
| 11 | FRIS-компактность групп пациентов с артериальной гипертензией                                                                      | 519.237.07            | 519.7                     |
| 12 | Квантование массы и группа Лоренца                                                                                                 | 512.815.8             | 512.8                     |

1. Расширение и пополнение базы текстов. А именно — постоянные пополнения базы данных из обновляющейся библиотеки, например, [elibrary.ru](http://elibrary.ru) или CyberLeninka с полноценной серверной частью, работающей без вмешательства пользователя.
2. Построение более высокоуровневой модели «обучения с учителем» на основе полученных данных от метода  $k$ -средних.

Разработанное программное обеспечение передано в редакцию журнала «Математические структуры и моделирование» для использования в работе редакции и сбора отклика пользователей с целью дальнейшего совершенствования ПО.

### Благодарности

Выражаем признательность Богаченко Надежде Фёдоровне и Вишняковой Ольге Анатольевне за внимание, проявленное к нашей работе.

### ЛИТЕРАТУРА

1. Sokirko A. A short description of Dialing Project. 2001. URL: <http://www.aot.ru/docs/sokirko/sokirko-candid-eng.html> (дата обращения: 28.08.2017).
2. Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees. URL: <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf> (дата обращения: 28.08.2017).
3. Steinhaus H. Sur la division des corps materiels en parties. Bull. Acad. Polon. Sci., 1956. P. 801–804.
4. Lloyd S. Least square quantization in PCM's. Bell Telephone Laboratories Paper, 1957. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.1338&rep=rep1&type=pdf> (дата обращения: 28.08.2017).
5. Gordon A.D. Classification, 2nd Edition. CRC Press, 1999. 272 p.

### APPLICATION OF CLUSTER ANALYSIS BY THE K-MEANS METHOD FOR THE CLASSIFICATION OF SCIENTIFIC TEXTS

**U.A. Osipova**

Student, e-mail: [fray09@inbox.ru](mailto:fray09@inbox.ru)

**D.N. Lavrov**

Ph.D. (Eng.), Associate Professor, e-mail: [dmitry.lavrov72@gmail.com](mailto:dmitry.lavrov72@gmail.com)

Dostoevsky Omsk State University

**Abstract.** The article discusses the construction of software to develop recommendations for the appointment of UDC. The development is based on the k-means method.

**Keywords:** method of k-means, lemmatization, TF-IDF.

*Дата поступления в редакцию: 29.06.2017*

## **ИССЛЕДОВАНИЕ СПОСОБОВ СОКРЫТИЯ ИНФОРМАЦИИ В ВИДЕОПОТОКЕ MPEG**

**Д.М. Бречка**

к.т.н., доцент кафедры компьютерных технологий и сетей,  
e-mail: dbrechkawork@yandex.ru

**А.А. Литвиненко**

студент, e-mail: a.litvinenko@outlook.com

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** Рассматриваются способы сокрытия информации в видеопотоке MPEG-4 с использованием кодека H.264. Для проверки возможности встраивания информации в видеопоток был выбран метод встраивания в коэффициенты дискретного косинусного преобразования. На основе кодека OpenH264 была реализована возможность встраивать и извлекать информацию.

**Ключевые слова:** кодек, MPEG-4, H.264, встраивание данных, видеопоток, сокрытие информации.

### **Введение**

Технологии сокрытия информации — это методики, позволяющие поместить определённую информацию внутри другой цифровой информации (такой как текст, изображения, звук и т. д.). Основные области применения для технологий сокрытия информации — это обеспечение конфиденциальности передаваемой информации, защита авторских прав.

В зависимости от методов реализации и целей применения все технологии сокрытия информации делятся на технологии сокрытия данных и технологии сокрытия цифровых водяных знаков. Сокрытие данных оперирует двумя наборами данных — носителем данных («обложкой») и встраиваемыми данными («сообщением»). Процесс сокрытия данных — это процесс встраивания «сообщения» в цифровую «обложку».

Целью данной работы является исследование процессов сокрытия и восстановления информации в видеопотоке MPEG-4 с использованием существующей реализации кодека H.264 с открытым исходным кодом.

## 1. Исследование стандарта MPEG-4

### 1.1. MPEG-4 и H.264

MPEG-4 — это формат медиаконтейнера, описанный в стандарте MPEG-4 Part 14 [1]. Медиаконтейнер MPEG-4 может использоваться как для хранения видео- и аудиопотока, так и для хранения других данных, например, субтитров и статичных изображений. Данные добавляются в контейнер в виде независимых потоков.

Самые распространённые форматы видео, поддерживаемые в MPEG-4, это:

- MPEG-4 Part 10 AVC/H.264;
- MPEG-4 Part 2 ASP;
- DivX/XviD.

Видеокодек H.264 в настоящее время является самым распространённым кодеком, используемым для кодирования видео в формате MPEG-4 [2]. Данный кодек позволяет гибко конфигурировать качество результирующего видео, предоставляя выбор между сжатием без потерь и сжатием с потерями.

### 1.2. Цветовое пространство

Видео в формате H.264 кодируется в цветовом пространстве YCbCr. Для представления каждого цвета YCbCr использует три значения: яркость (Y) и две цветоразностные компоненты — синюю (Cb) и красную (Cr). Само по себе цветовое пространство, по сути, является способом представления цветового пространства RGB и ничем принципиально от него не отличается. Но в отличие от RGB, YCbCr позволяет использовать особенность человеческого зрения в своих целях, произведя субдискретизацию цветоразностных компонент. Дело в том, что человеческий глаз воспринимает изменения в яркости сильнее, чем изменения в цветности. Использование YCbCr позволяет передавать информацию о яркости с полным, а цветоразностную информацию — с уменьшенным разрешением.

Структура дискретизации обозначается как соотношение  $X : a : b$ , где  $X$  — частота дискретизации яркостного сигнала (она же ширина макропикселя),  $a$  и  $b$  — число выборок цветоразностных сигналов в первой и второй строках соответственно. В формате H.264 используется структура 4:2:0, что позволяет сократить горизонтальное и вертикальное цветовые разрешения вдвое, но при этом сохранить яркостное разрешение исходным.

### 1.3. Структура кодека

Видеокодек представляет входную видеопоследовательность (или набор изображений) в сжатом виде и декодирует её таким образом, чтобы создать точную или приближенную копию входной последовательности. В случае, если копия получается идентичной оригиналу, такое кодирование называется кодированием без потерь (loseless), иначе — кодированием с потерями (lossy) [2].



Видеокодек представляет исходную последовательность в виде модели — эффективно закодированного представления, которое затем может использоваться для восстановления приближенной копии исходных данных. В идеальных условиях модель должна представлять исходную последовательность как можно меньшим количеством бит, но при этом обеспечивать как можно большую точность.

Кодировщик видео состоит из трёх основных функциональных блоков — временной модели, пространственной модели и энтропийного кодировщика. На вход временной модели подаются несжатые видеоданные или набор кадров. Временная модель пытается сократить временную избыточность, используя повторяющиеся участки в соседних кадрах. Обычно это делается путём прогнозирования кадра из предыдущего. В H.264 прогноз формируется из одного или нескольких предыдущих или следующих кадров и затем улучшается путём компенсации разницы между кадрами (так называемый прогноз с компенсацией движения). На выход временной модели попадают кадр разницы, получаемый путём вычитания прогноза из текущего кадра, и набор параметров — векторов движения, описывающих то, как именно было компенсировано движение.

Полученный кадр разницы затем поступает на вход пространственной модели, которая анализирует схожесть соседних областей в кадре и использует её для уменьшения пространственной избыточности. В H.264 для этих целей используется трансформация кадров разницы и последующее квантование полученных значений. В процессе трансформации сэмплы преобразуются и представляются в виде набора коэффициентов трансформации, которые затем подвергаются квантованию, что позволяет оставить только малое число наиболее значимых коэффициентов и представить кадр разницы в более компактном виде. В результате на выход пространственной модели попадает набор коэффициентов трансформации.

Полученные параметры временной модели (вектора движения) и параметры пространственной модели (коэффициенты трансформации) затем сжимаются энтропийным кодировщиком. Задача энтропийного кодировщика — устранить статистическую избыточность в полученных на вход данных. Например, энтропийное кодирование может заменить наиболее часто встречающиеся вектора движения на короткие двоичные коды. На выход энтропийного кодировщика поступает сжатый двоичный поток, который затем можно сохранить в файл или передать по сети. Сжатая последовательность состоит из закодированных параметров векторов движения, закодированных коэффициентов трансформации и заголовка.

Декодер видео восстанавливает кадры из сжатого потока данных. Коэффициенты трансформации и вектора движения декодируются энтропийным декодером, после чего пространственная модель восстанавливается до кадра разности. Декодер использует параметры векторов движения в связке с одним или более ранее декодированными кадрами, создавая прогноз для текущего кадра. Итоговый кадр реконструируется путём добавления кадра разности к прогнозу.

Схематично структура кодкека представлена на рисунке 1.



Рис. 1. Структура кодека H.264

#### 1.4. Временная модель

Макроблок, представляющий собой участок кадра размером  $16 \times 16$  пикселей, является базовой единицей для прогнозирования с компенсацией движения в целом ряде стандартов кодирования видео, включая MPEG-1, MPEG-2, H.261, H.263 и H.264. Для входных данных в формате 4:2:0 макроблок размером  $16 \times 16$  пикселей представляет собой совокупность четырёх блоков яркости размером  $8 \times 8$  каждый, двух блоков цветности размером  $8 \times 8$  для синей и красной цветоразностных компонент соответственно. Кодек H.264 разбивает каждый кадр на макроблоки и затем оперирует ими по отдельности. Для каждого макроблока размером  $16 \times 16$  выполняются следующие действия.

1. Из ранее закодированного и переданного кадра берутся блоки размером  $M \times N$  и выполняется поиск похожих блоков в текущем кадре, для этого блок из предыдущего кадра сравнивается со всеми возможными блоками такого же размера в текущем кадре и выбирается наиболее похожий. Операция сравнения вычитает блоки-кандидаты из исходного блока и получает блоки разницы. Блок-кандидат, которому соответствует блок разницы, содержащий наименьшее количество данных, считается наилучшим совпадением. Процесс поиска наилучшего совпадения называется оценкой движения (motion estimation).
2. Найденное наилучшее совпадение становится прогнозом для текущего блока  $M \times N$  и далее используется соответствующий ему блок разницы. Формирование блока разницы путём вычитания прогнозируемого блока из текущего блока называется компенсацией движения (motion compensation).
3. Полученный блок разницы кодируется и передаётся вместе со значениями смещения прогнозируемого блока относительно текущего блока.

Декодер, в свою очередь, получает на вход вектор движения для вычисления прогнозируемой области, декодирует блок разности, складывает его с блоком-предшественником и получает версию текущего блока.

В случаях, когда текущий кадр отличается от предыдущего достаточно сильно, может быть более логичным закодировать макроблок без компенсации движения. В результате кодировщик для каждого макроблока может выбирать между внутрирегиональным режимом (intra, кодирование без компенсации движений) и межрегиональным режимом (inter, кодирование с компенсацией движений).

Движущиеся объекты редко соответствуют границам блоков  $16 \times 16$ , и для достижения лучших показателей более удобным было бы использовать блоки разных размеров. Объекты могут менять своё положение не на целое число пикселей, поэтому при поиске региона-кандидата можно использовать не только регионы с целочисленными координатами, но и субпиксельные.

При уменьшении размеров блоков разности ( $16 \times 16$ ,  $8 \times 8$ ,  $4 \times 4$ ) улучшается качество изображения, но увеличиваются сложность (необходимо выполнить большее число операций для поиска блока-кандидата) и количество данных (количество векторов движения, которые необходимо передать). Компромиссом является выбор размера блока в зависимости от характеристик изображения — блоки большего размера для однотонных регионов и меньшие размеры блоков для регионов с большим количеством деталей. Н.264 использует блоки варьирующихся размеров для компенсации движения.

Для поиска наиболее точного совпадения в субпиксельных координатах кодек Н.264 может сперва найти наиболее точное совпадение в целочисленных координатах, затем выполнить поиск рядом с этим регионом в полупиксельных координатах, а затем и в четвертьпиксельных, если это позволит увеличить точность результата.

### 1.5. Модель изображения

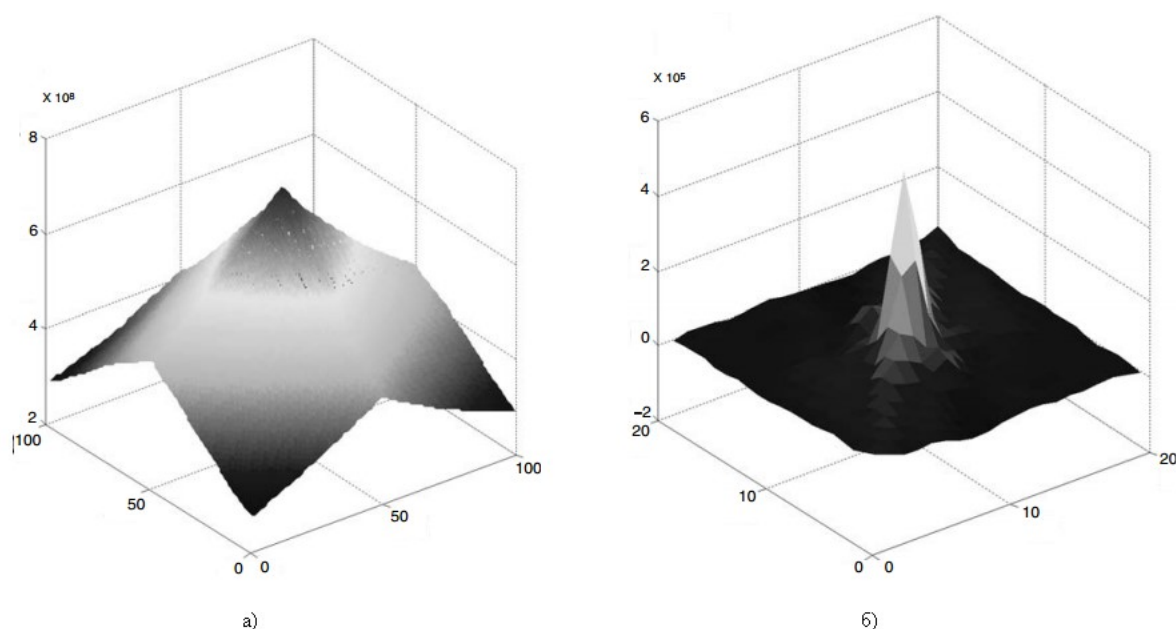


Рис. 2. Двумерные автокорреляционные функции изображения и разницы

На рисунке 2а изображён график двумерной автокорреляционной функции для видеоизображения, на котором высота каждой точки выражает разницу между оригинальным изображением и его пространственно-смещённой копией.

Пик в центре графика соответствует нулевому смещению изображения. С удалением пространственно-смещённой копии от центра оригинального изображения в любом направлении значение функции постепенно уменьшается, образуя на графике постепенный наклон, что свидетельствует о высокой корреляции соседних значений.

На 26 изображён график двумерной автокорреляционной функции для разностного видеоизображения, значения функции на котором стремительно уменьшаются с увеличением смещения, что свидетельствует о низкой корреляции соседних значений. При компенсации движений, уменьшающей корреляцию соседних значений в разнице, сжатие разницы происходит быстрее и эффективнее, чем сжатие исходного видеоизображения. Функция модели изображения состоит в декорреляции изображения или вычета и преобразовании его в форму, которая затем может быть легко сжата энтропийным кодировщиком. Модель изображения состоит из 3 частей — трансформации (декоррелирует и уменьшает размер данных), квантования (уменьшает точность трансформированных данных) и переупорядочивания (выстраивает данные, группируя значимые данные вместе).

### 1.6. Кодирование с предсказанием

Компенсация движения является частным случаем кодирования с предсказанием, при котором кодировщик формирует предсказание для региона в текущем кадре на основе данных из предыдущих кадров и вычитает прогноз из текущего региона для формирования разницы. Если предсказание успешно, то объём данных, занимаемых разницей, получается меньше объёма фрейма и разница представляется меньшим количеством бит. Подобным же образом участок изображения может быть представлен с помощью других, закодированных ранее, участков в этом же кадре. Кодирование с предсказанием используется для внутрикадрового кодирования в H.264.

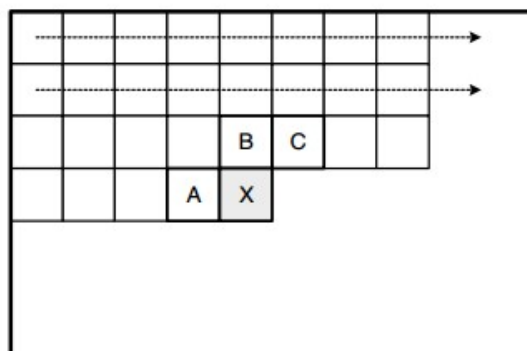


Рис. 3. Пространственное предсказание

Рисунок 3 показывает, как кодируется изображение при пространственном предсказании. Буквой X обозначен пиксель, который необходимо зако-

дировать в данный момент,  $A$ ,  $B$  и  $C$  — соседние пиксели, закодированные/декодированные ранее. Кодировщик вычисляет прогноз для  $X$ , основываясь на ранее закодированных пикселях  $A$ ,  $B$  и  $C$ , вычитает предсказание из  $X$  и возвращает разницу. Декодер в свою очередь тоже вычисляет прогноз на основе ранее декодированных пикселей  $A$ ,  $B$  и  $C$  и складывает полученное предсказание и разницу, получая на выходе исходный пиксель  $X$ .

### 1.7. Кодирование с преобразованием

Целью шага кодирования с преобразованием является представление изображения или разницы в ином виде (домене преобразования). Выбор метода трансформации может зависеть от ряда критериев.

1. Данные в домене преобразования не должны коррелировать (разделены на части со слабой внутренней связностью) и быть компактными (большая часть преобразованных данных должна содержаться в малом количестве значений).
2. Преобразование должно быть обратимым.
3. Преобразование должно быть как можно менее ресурсоёмким.

В случае с H.264 эту роль могут выполнять дискретное косинусное преобразование или дискретное вейвлет-преобразование. Дискретное косинусное преобразование лучше подходит для использования при компенсации движения на основе блоков, так как само основано на блоках, в отличие от дискретного вейвлет-преобразования, которое предназначено для обработки изображений целиком.

### 1.8. Дискретное косинусное преобразование

Дискретное косинусное преобразование оперирует блоками изображения или блоками вычетов  $X$  размера  $N \times N$  и подаёт на выход блоки коэффициентов  $Y$  такого же размера. Преобразование  $X$  в  $Y$  осуществляется по формуле

$$Y = AXA^T.$$

Обратное преобразование выполняется по формуле

$$X = A^T Y A,$$

где  $X$  — матрица исходных значений,  $Y$  — матрица коэффициентов и  $A$  — матрица преобразования размером  $N \times N$ , элементы которой рассчитываются по формуле:

$$A_{ij} = C_i \cos \frac{(2j+1)i\pi}{2N}, C_i = \begin{cases} \sqrt{\frac{1}{N}}, & \text{если } i = 0; \\ \sqrt{\frac{2}{N}}, & \text{если } i > 0 \end{cases}$$

В результате дискретного косинусного преобразования  $N^2$  значений из исходной матрицы преобразуются в  $N^2$  коэффициентов. Затем можно сохранить

только наиболее значимые коэффициенты, а наименее значимые коэффициенты отбросить. Изменяя количество хранимых коэффициентов, можно управлять качеством итогового видео.

### 1.9. Квантование

Квантователь преобразует сигнал из множества значений  $X$  в множество значений  $Y$ . Квантование позволяет представить исходный сигнал в виде меньшего числа бит, так как диапазон выходных значений меньше диапазона входных. Скалярный квантователь преобразует единицу входного сигнала в единицу выходного, а векторный квантователь преобразует группу входных сигналов в группу выходных.

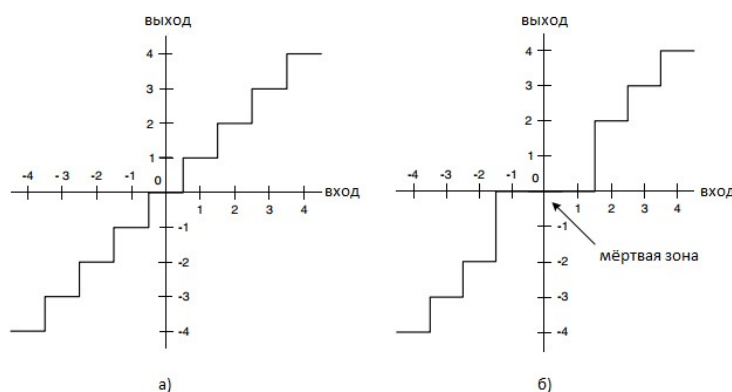


Рис. 4. Скалярные квантователи: а) линейный и б) нелинейный с мёртвой зоной

Рисунок 4 демонстрирует два частных случая скалярного квантователя — линейный квантователь (с прямым соотношением входных и выходных значений) и нелинейный квантователь с мёртвой зоной в области 0 (при котором значения, близкие к 0, переходят в значение 0).

В видеокодеках квантователь обычно состоит из двух частей — прямого квантователя (FQ) в кодировщике и обратного квантователя (IQ) в декодере. Критичным параметром является размер шага (QP). Большой размер шага позволяет уменьшить диапазон квантованных значений, а значит, и уменьшить объём передаваемых данных. Но это повлечёт за собой ухудшение качества, так как восстановленные декодером (обратным квантователем) значения будут являться грубым приближением. В случае с меньшим размером шага квантованные данные будут более приближенными к оригиналу, но увеличат объём передаваемых данных и негативно повлияют на эффективность сжатия.

В H.264 квантование используется для отсекаания околонулевых малозначимых коэффициентов, полученных в результате дискретного косинусного преобразования. На вход прямого квантователя поступают наборы коэффициентов ДКП, а на выход — разрежённый массив квантованных коэффициентов, большая часть значений в котором представлена нулями.

Векторный квантователь сопоставляет последовательность (вектор) входных данных (такую, как блок изображения) в виде одного значения (кодowego слова), а декодер в свою очередь сопоставляет каждое кодowego слово приближенному значению оригинальной входной последовательности. В общем случае процесс векторного квантования состоит из следующих шагов:

- 1) входное изображение разбивается на блоки размером  $M \times N$ ;
- 2) из кодовой книги выбирается вектор, наиболее похожий на текущий;
- 3) на выход передаётся индекс найденного кодowego слова;
- 4) декодер восстанавливает приближенную копию исходного вектора (и, соответственно, исходного блока), используя кодовую книгу и индекс вектора.

Основными проблемами в данной ситуации являются составление кодовой книги таким образом, чтобы обеспечить наилучшее качество восстановленного изображения, и скорость поиска наиболее подходящего вектора в кодовой книге.

### 1.10. Переупорядочивание

Квантованные коэффициенты трансформации должны быть максимально сжаты перед дальнейшим хранением и передачей. Так как данные, полученные после выполнения квантования, содержат в себе большое число нулевых и малое число ненулевых значений, то имеет смысл произвести переупорядочивание квантованных коэффициентов (сгруппировать ненулевые коэффициенты вместе) и представить нулевые коэффициенты в более компактном виде. Эти шаги необходимо сделать перед выполнением энтропийного кодирования.

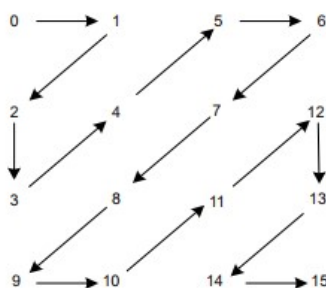


Рис. 5. Сканирование зигзагом

Оптимальный путь для переупорядочивания зависит от распределения ненулевых коэффициентов дискретного косинусного преобразования ДКП. Для типичного блока кадра путь переупорядочивания будет представлять собой зигзаг (рисунок 5), начинающийся из левого верхнего пикселя. Все коэффициенты ДКП, следуя по пути переупорядочивания, помещаются в одномерный массив. В результате переупорядочивания на выход поступает массив, обычно состоящий из 1 или более кластеров ненулевых коэффициентов в начале,

за которыми следует последовательность нулей. Последовательность ненулевых коэффициентов может быть представлена двумя способами — двумерным кодированием пробег-уровень или трёхмерным кодированием пробег-уровень. В первом случае каждый ненулевой коэффициент представляется в виде пары чисел (пробег, уровень), где пробег — это количество нулей, предшествующих ненулевому значению, а уровень — непосредственное значение коэффициента. Для обозначения того, что ненулевые коэффициенты закончились и далее следуют только нули, в выходную последовательность вставляется специальный символ, обозначающий конец последовательности. Во втором случае каждый элемент последовательности кодируется тремя значениями (пробег, уровень, последний), где пробег и уровень соответствуют аналогичным значениям в первом случае, а последний — признак конца последовательности (0, если далее следует ещё один ненулевой коэффициент, и 1, если данный ненулевой коэффициент — последний).

### 1.11. Энтропийный кодировщик

Энтропийное кодирование преобразует набор значений, представляющих элементы видеопоследовательности, в сжатый поток бит, подходящий для хранения или передачи. Наборы значений могут включать в себя квантованные коэффициенты трансформации, вектора движения, заголовки макроблоков и изображений и дополнительную информацию.

Некоторые элементы видеопоследовательности могут довольно сильно коррелировать друг с другом, например, вектора движения или коэффициенты ДКП в соседних блоках. Эффективность сжатия может быть улучшена, если кодировать не отдельный блок или макроблок, а разницу, зависящую от реальных и ранее закодированных данных.

Вектор движения блока или макроблока означает смещение относительно связанного блока из предыдущего кадра. Вектора соседних блоков довольно часто коррелируют, так как движущиеся объекты могут занимать большие участки кадра. Это утверждение наиболее справедливо для блоков малых размеров ( $4 \times 4$ ) и больших движущихся объектов. Сжатие векторов движения может быть выполнено путём вычисления зависимости вектора от ранее закодированных векторов. Разница между реальным вектором и спрогнозированным значением затем кодируется и передаётся вместо оригинального значения.

Кодировщик с переменной длиной преобразует набор входных значений в набор кодовых слов переменной длины. При этом часто встречающиеся значения представляются в виде кодовых слов меньшей длины. Кодирование Хаффмана сопоставляет каждое входное значение с кодовым словом переменной длины, основываясь на частоте вхождения разных входных значений. Согласно оригинальной схеме, предложенной Хаффманом в 1952 году, перед кодированием необходимо сперва рассчитать частоту вхождений всех входных значений и построить набор кодов переменной длины. Если вероятностное распределение рассчитано верно, код Хаффмана позволяет представить входные данные в сравнительно более компактной форме.



Код Хаффмана имеет два недостатка при его практическом применении в работе видеокодека. Во-первых, декодер должен использовать тот же набор кодовых слов, что использовал кодировщик. Передача таблицы соответствия кодов Хаффмана исходным данным требует дополнительных расходов и снижает эффективность сжатия, особенно для видеопоследовательностей малой длины. Во-вторых, таблица вероятностей для видеопоследовательности большой длины не может быть построена до тех пор, пока не закодирован весь видеопоток, что может повлечь за собой большие задержки в процессе кодирования. Для решения этой проблемы H.264 предлагает заранее рассчитанные таблицы вероятностей, созданные на основе некоторого стандартного видеоматериала.

Стандарт MPEG-4 позволяет представить трёхмерные коды квантованных коэффициентов (пробег, уровень, последний) как одно из 102 кодовых слов переменной длины, таблица которых определена в стандарте. В случае, если кодового слова для нужного трёхмерного кода нет в таблице, то трёхмерный код представляется в виде 13-битного кода, содержащего все три значения, и записывается после управляющей последовательности (0000011). Векторы движения представляются парой кодовых слов переменной длины (одно значение для координаты  $x$  и одно — для  $y$ ). Стандарт MPEG-4 также предлагает для этих целей готовую таблицу соответствия.

## 2. Исследование способов сокрытия данных в H.264

### 2.1. Сокрытие данных в векторах движения

Одним из наиболее очевидных способов сокрытия данных в видеопотоке является сокрытие в векторах движения [3,4]. Для уменьшения влияния скрытых данных на итоговое видео данные записываются только в младший бит обеих координат вектора. Качество итогового видео тем лучше, чем большая точность вектора используется при кодировании. Полу- или четвертьпиксельная точность вектора движения позволяет скрывать данные практически без ущерба для итогового изображения.

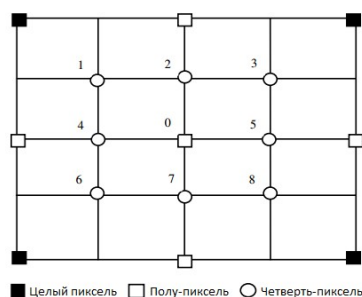


Рис. 6. Нумерация пикселей

Частным случаем является использование свойства чётности координат вектора для сокрытия данных. При таком подходе для хранения координат вектора

используется четвертьпиксельная точность вектора (рисунок 6), а все возможные точки поиска разбиваются на 2 группы — (0, 1, 3, 6, 8) и (2, 4, 5, 7). Каждой из двух групп точек ставится в соответствие бит 0 или 1. Соккрытие данных происходит в процессе оценки движения. После поиска наилучшего полупиксельного совпадения все соседние четвертьпиксели разбиваются на группы и в каждой группе выбирается наилучший кандидат. Затем выбирается группа, соответствующая значению, которое необходимо скрыть (0 или 1). Наилучший кандидат из соответствующей группы считается итоговым наилучшим кандидатом.

## 2.2. Соккрытие данных в макроблоках

Ещё одним методом является соккрытие данных в макроблоках [5], а точнее, в признаке принадлежности макроблоков к определённым группам слайсов. Кадр в H.264 состоит из макроблоков, которые затем могут быть разбиты на слайсы, а слайсы объединены в группы слайсов. Каждая группа слайсов состоит из одного или более слайса и макроблоков, расположенных в любом порядке. Соответствие макроблоков группам слайсов может использоваться для соккрытия данных в видеопотоке. Так как макроблоки могут быть произвольным образом сопоставлены с группой слайсов, то биты данных могут быть скрыты в поле идентификатора группы слайсов в макроблоке.

В случае, если существуют две группы слайсов, назначение макроблока в первую группу слайсов может кодировать бит 0, а назначение во вторую — бит 1. Таким образом, каждый макроблок может скрывать 1 бит информации. Так как стандарт H.264 позволяет использовать максимум 8 групп слайсов на кадр, количество кодируемых значений можно увеличить до 8, тем самым позволяя в одном макроблоке скрывать до 3 бит информации (таблица 1).

Таблица 1. Соответствие числа групп слайсов количеству возможных скрытых бит

| Количество групп слайсов в кадре | Возможные скрытые значения в макроблоке | Скрытых бит в макроблоке |
|----------------------------------|-----------------------------------------|--------------------------|
| 2                                | 0; 1                                    | 1                        |
| 4                                | 00; 01; 10; 11                          | 2                        |
| 8                                | 000; 001; 010; 011; 100; 101; 110; 111  | 3                        |

Если кадр содержит  $m$  макроблоков, то в кадре можно скрыть  $m \times n$  бит информации. Процесс кодирования выполняется следующим образом. В первую очередь считываются  $n \times t$  бит скрываемых данных, где  $n$  — значение 1, 2 или 3, согласно таблице 1. Затем каждому макроблоку в кадре по очереди ставится в соответствие нужная группа слайсов в зависимости от прочитанных скрываемых данных. Процесс декодирования выглядит соответствующим образом — для каждого макроблока в кадре берётся значение поля идентификатора группы слайсов, значение этого поля является частью скрытого сообщения.

### 2.3. Сокрытие данных в квантованных коэффициентах ДКП

Также существует способ сокрытия данных в коэффициентах, полученных в результате дискретного косинусного преобразования и последующего квантования [6]. При данном подходе сначала выбирается некое граничное значение от 0 до 15, определяющее максимально допустимый номер коэффициента ДКП, разрешённого для модификации. Выбор граничного значения зависит от количества информации, которую необходимо скрыть. Кодирование осуществляется путём перебора всех блоков разницы и выбора из них таких блоков, для которых последний ненулевой коэффициент имеет индекс не меньше, чем граничное значение. Если подходящий коэффициент ( $V_a$ ) найден, то он изменяется в зависимости от того, какое значение ( $D_a$ ) необходимо скрыть:

Если  $D_a = 1$ :

$$D_a = \begin{cases} V_a, & \text{если } V_a \% 2 = 1; \\ V_a - 1, & \text{если } V_a \% 2 = 0 \text{ и } V_a > 0; \\ V_a + 1, & \text{если } V_a \% 2 = 0 \text{ и } V_a < 0. \end{cases}$$

Если  $D_a = 0$ :

$$D_a = \begin{cases} V_a, & \text{если } V_a \% 2 = 0; \\ V_a - 1, & \text{если } V_a \% 2 = 1 \text{ и } V_a > 0; \\ V_a + 1, & \text{если } V_a \% 2 = 1 \text{ и } V_a < 0. \end{cases}$$

То есть если встраиваемое значение — 1, то значение коэффициента будет нечётным, а если 0 — то чётным.

Процесс декодирования выглядит следующим образом — сначала из всех наборов коэффициентов выбираются те, у которых индекс последнего ненулевого коэффициента не меньше, чем граничное значение (аналогично процессу кодирования). Из каждого подходящего набора коэффициентов выбирается последний ненулевой ( $V_a$ ) и выполняется извлечение скрытого бита данных ( $D_a$ ) по следующему принципу:

$$D_a = \begin{cases} 1, & \text{если } V_a \% 2 = 1; \\ 0, & \text{если } V_a \% 2 = 0. \end{cases}$$

Преимуществом такого подхода является простота сокрытия данных. Так как данный алгоритм использует признак чётности для сокрытия данных и вероятности того, что значение последнего ненулевого коэффициента, удовлетворяющего условию, будет чётным или нечётным, равны и составляют 50%, то есть 50-процентная вероятность того, что коэффициент останется неизменным после сокрытия данных. Это положительно сказывается как на качестве результирующего видео, так и на скорости кодирования. Скорость кодирования видео при использовании этого метода практически не изменяется, а декодирование данных происходит без каких-либо потерь.

### 3. Экспериментальное исследование возможности сокрытия данных

#### 3.1. Подготовка эксперимента

Из рассмотренных методов сокрытия информации было решено выбрать сокрытие в коэффициентах дискретного косинусного преобразования. Так как реализация собственного кодировщика и декодировщика является очень сложной задачей, было решено модифицировать уже готовое решение. Большинство решений не предоставляют свои исходные коды для модификации и не позволяют вмешиваться в процесс своей работы.

Были найдены две открытые реализации кодека H.264 — одна от Института Генриха Герца [7], а вторая от компании Cisco [8]. Выбор пал на вторую как наиболее надёжную, активно разрабатываемую и поддерживаемую (последняя стабильная версия опубликована 11 июля 2016). Данный кодек написан на языке C++ с ассемблерными вставками.

Кодировщик в качестве параметра командной строки принимает путь к файлу конфигурации, в котором описываются все настраиваемые параметры, такие как количество кадров в секунду и размер кадра, а также пути к входному и выходному файлам. Входной файл должен представлять собой набор сырых данных в формате YUV (*YCbCr*), а в выходной файл будет сохранено видео в формате H.264 без контейнера.

Декодер в качестве параметров командной строки позволяет указать пути к входному и выходному файлам. Здесь в качестве входного файла должен выступать файл с H.264 видео без контейнера, а в качестве выходного будет создан файл с сырыми YUV данными. Для получения сырого H.264 видео из файла с видео, использующего контейнер MPEG-4, можно использовать утилиту *ffmpeg* [9–11].

#### 3.2. Модификация кодека

Модификация кодека будет заключаться в добавлении программного кода, позволяющего считывать встраиваемые данные из файла, встраивающего данные в видеофайл, извлекающего данные из видеофайла и записывающего извлечённые данные в файл.

Для начала необходимо реализовать методы, осуществляющие чтение данных из файла и возвращающие значение следующего бита данных.

1. В файл *walsenc.cpp* добавим метод *ReadPayloadFile* для чтения файла с данными.
2. В этом же файле в структуру *SFilesSet* добавим строковую переменную *strPlFile*, а в метод *ParseConfig* добавим чтение параметра из файла конфигурации.
3. Добавим новый объект типа *FILE\** с именем *pFpPl* и код открытия файла на чтение и соответствующие проверки.

4. Добавим вызов метода *ReadPayloadFile* и возвращенное им значение передадим в вызов метода *EncodeFrame* в качестве второго аргумента.
5. Изменим вызов и объявление метода *WelsEncRec16x16Y*, добавляя массив со скрываемыми данными в качестве последнего аргумента.
6. В самый конец метода *WelsEncRec16x16Y* добавим логику, непосредственно связанную с сокрытием информации.

Теперь необходимо реализовать алгоритм восстановления данных в декодере.

1. Сначала необходимо модифицировать разбор параметров командной строки. Для этого в метод *main* файла *h264dec.cpp* добавим строковую переменную *strPayloadFile* и модифицируем код так, чтобы значение переменной бралось из третьего аргумента командной строки.
2. Модифицируем вызов и описание метода *H264DecodeInstance*, добавив путь к файлу с восстановленными данными в качестве последнего параметра.
3. В метод *H264DecodeInstance* добавим код, осуществляющий открытие файла для хранения восстановленных данных на запись, создание массива для хранения декодированных данных и запись получившихся данных в файл.
4. Изменим объявление и вызов метода *DecodeFrameNoDelay*, добавив в качестве последнего аргумента параметр типа *char\** для хранения восстановленных данных.
5. Восстановление данных можно выполнить внутри метода *Rec16x16Mb*, изменив его объявление и вызов, добавив в качестве последнего аргумента файл восстановленных данных, как и в предыдущем случае.
6. В самый конец метода *Rec16x16Mb* добавим логику, непосредственно ответственную за восстановление данных.

### 3.3. Проверка корректности работы

Для проверки корректности работы необходимо выполнить следующие шаги.

1. Перекомпилируем кодек.
2. Создадим файл с данными, которые необходимо скрыть:  
*echo "Data hiding in H.264 video. -OmSU, 2017" > payload.txt*
3. Создадим файл конфигурации. В качестве основы можно взять файл *welsenc\_vd\_1d.cfg*, поставляющийся с кодеком, указать в нём следующие параметры:
  - *InputFile* — имя входного файла (*source.yuv*);
  - *OutputFile* — имя выходного файла (*test.h264*);
  - *PayloadFile* — имя файла со скрываемыми данными (*payload.txt*);
  - *SourceWidth* — ширина изображения в исходном видео (1920);
  - *SourceHeight* — высота изображения в исходном видео (1080).
4. Запустим кодировщик, передав ему файл конфигурации в качестве единственного параметра:

*h264enc payload.cfg*

5. Запустим декодер со следующими параметрами:

*../h264dec test.h264 test.yuv payload.new*

6. Сравним исходные и полученные данные.

Результат эксперимента показал, что восстановленный файл содержит все данные из оригинального файла, что свидетельствует о правильной работе кодека и реализованного алгоритма.

## Заключение

В данной работе были рассмотрены основные способы сокрытия информации в видеопотоке H.264. Для проверки возможности встраивания информации в видеопоток был выбран метод встраивания в коэффициенты ДКП. На основе кодека OpenH264 была реализована возможность встраивать и извлекать информацию. Результаты эксперимента показали, что выбранная схема применима для решения поставленной задачи.

## ЛИТЕРАТУРА

1. Information technology – Coding of audio-visual objects – Part 14: MP4 file format. International Standard, 2003.
2. Richardson I.E.G. H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia. Wiley, 2003.
3. Li Y., Chen H.-X., Zhao Y. A new method of data hiding based on H.264 encoded video sequences // IEEE 10th International Conference on Signal Processing (ICSP). 2010. P. 1833–1836.
4. Aly H.A. Data hiding in motion vectors of compressed video based on their associated prediction error // IEEE Trans. Inform. Forensics Security. 2011. V. 6, № 1. P. 14–18.
5. Lathikanandini M., Suresh J. Steganography in MPEG Video Files using MACROBLOCKS // IEEE Trans. Inform. Forensics Security. 2012. V. 3, № 1.
6. Wong K., Tanaka K., Takagi K., Nakajima Y. Complete video quality-preserving data hiding // IEEE Transactions on Circuits and Systems for Video Technology. 2009. V. 19, № 10.
7. H.264/MPEG4-AVC URL: <https://www.hhi.fraunhofer.de/index.php?id=468&L=1> (дата обращения: 15.03.2017).
8. OpenH264 URL: <https://www.openh264.org/> (дата обращения: 15.03.2017).
9. H.264/MPEG-4 AVC Video Compression Tutorial. URL: [http://web.cs.ucla.edu/classes/fall103/cs218/paper/H.264\\_MPEG4\\_Tutorial.pdf](http://web.cs.ucla.edu/classes/fall103/cs218/paper/H.264_MPEG4_Tutorial.pdf) (дата обращения: 23.12.2016).
10. Extracting h264 raw video stream from mp4 or flv with ffmpeg generate an invalid stream. URL: <http://stackoverflow.com/questions/19300350/extracting-h264-raw-video-stream-from-mp4-or-flv-with-ffmpeg-generate-an-invalid> (дата обращения: 23.12.2016).
11. How to convert raw H.264 stream from video call to mp4 file. URL: <http://stackoverflow.com/questions/29293836/>

how-to-convert-raw-h-264-stream-from-video-call-to-mp4-file  
(дата обращения: 23.12.2016).

## INVESTIGATION OF METHODS OF INFORMATION HIDING IN THE MPEG VIDEO STREAM

**D.M. Brechka**

Ph.D.(Eng.), Associate Professor, e-mail: dbrechkawork@yandex.ru

**A.A. Litvinenko**

Student, e-mail: a.litvinenko@outlook.com

Dostoevsky Omsk State University

**Abstract.** The article discusses the methods of information hiding in the MPEG-4 video stream using the H.264 codec. To test the possibility of embedding of information in the video stream, there was chosen method of embedding in the coefficients of the discrete cosine transform. The ability of embedding and retrieving of information was tested on modified OpenH264 codec.

**Keywords:** codec, MPEG-4, H.264, embedding data, video stream, hiding information.

*Дата поступления в редакцию: 17.03.2017*

## **ПРОГРАММА, МОДЕЛИРУЮЩАЯ КОМПЬЮТЕРНУЮ СЕТЬ И СЕТЕВЫЕ АТАКИ**

**А.К. Гуц**

профессор, д.ф.-м.н., кафедра кибернетики, e-mail: guts@omsu.ru

**Е.П. Эннс**

студент, e-mail: jecka-gtx@mail.ru

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** Представляется компьютерная программа, которая имитирует работу компьютерной сети, а также позволяет продемонстрировать результат атак злоумышленников на сеть, использующих некоторые имеющиеся в сети уязвимости.

**Ключевые слова:** компьютерная сеть, сетевые атаки, имитация атак, компьютерная программа.

### **Введение**

Анализу моделей атак на компьютерные сети и проверке эффективности этих атак посвящается много работ. Очень часто для этого используются тестовые компьютерные сети [1], как правило, являющиеся реальными компьютерными сетями. Однако это же можно делать, создавая компьютерные программы, имитирующие реальные сети, атаки на них с демонстрацией последствий [2–4].

В данной статье представляется простая учебная компьютерная программа [5], которая моделирует работу компьютерной сети, а также позволяет продемонстрировать результат атак злоумышленников на сеть, использующих некоторые имеющиеся в сети уязвимости. Компьютер хакера и компьютеры сети представляются окнами на мониторе программы.

Основные задачи, которые необходимо было решить:

- 1) создание окна злоумышленника;
- 2) создание окон коммутатора и маршрутизатора, работающих по схожим принципам, что и реальные устройства;
- 3) создание окна сервера;
- 4) создание базового функционала окон хостов и описание способ конфигурирования моделируемой сети;
- 5) представление функционала окон хостов, в особенности – злоумышленника (хакера);
- 6) обеспечение возможности передачи сообщений между окнами программы;



7) демонстрация, что выбранные уязвимости в моделируемой сети присутствуют, а атаки с их использованием — действенны.

Уязвимость компьютерной системы — это недостаток в системе, её слабое место, используя которое можно намеренно нарушить целостность, т. е. отдельные элементы системы перестают функционировать и это приводит к неправильной её работе. Уязвимость может быть результатом ошибок программирования, недостатков, допущенных при проектировании системы, ненадёжных паролей, вирусов и т. д.

## **1. Моделирование**

Для того чтобы наглядно показать действие атак, использующих уязвимости компьютерных сетей, требуется создать соответствующую компьютерную программу, с помощью которой можно имитировать нужную атаку.

Опишем возможный принцип работы такой программы.

### **1.1. Окна**

Компьютер злоумышленника (хакера) и компьютеры сети в программе представляются окнами на мониторе запущенной программы.

В каждом окне (кроме окна коммутатора) имеется возможность назначить адрес, а в окна компьютеров (и злоумышленника также) — ещё и назначить шлюз.

Внешние адреса машин в программе — 100 и больше.

Это сделано для моделирования локальной (доменной) сети и внешней сети (интернет). В окно злоумышленника добавлена возможность отправить пакет, собранный из отдельных «кусочков», каждый из которых задаётся в отдельности.

Программа использует свою структуру сетевого пакета в отличие от реальных сетей. Это допущение позволяет увеличить наглядность и упростить моделирование.

В реальности для атаки часто требуется большое количество хостов, мы ограничимся двумя-тремя.

### **1.2. Отсылка сообщения хакером**

Злоумышленник направляет сообщение в сеть, сеть его принимает: имеем передачу сообщений между двумя окнами, представляющими два компьютера.

Чтобы реализовать возможность передачи сообщений между окнами программы, было решено использовать несколько встроенных классов, обеспечивающих клиентские подключения для сетевых служб, работающих с потоками данных.

В функции отправки создаётся экземпляр класса «TcpClient», сообщение переводится из строкового типа в байтовый массив, который уже с помощью

открывшегося потока передачи данных посылается на указанный порт. Затем поток закрывается, а экземпляр класса удаляется:

```
//создание нового TCP клиента
TcpClient client = new TcpClient("localhost Convert.ToInt32(port));
byte[] data = new byte[256];
//перевод строки в байтовый массив
byte[] msg = Encoding.UTF8.GetBytes(str);
int count = msg.Length <= 256 ? msg.Length : 256;
Array.Copy(msg, data, count);
//создание потока передачи данных
NetworkStream stream = client.GetStream();
//запись строки в поток
stream.Write(data, 0, 256);
//закрытие потока и удаление TCP клиента
stream.Close();
client.Close();
textBox3.Text += "Отправлено " + oppPort + »» " + str;
textBox3.Text += Environment.NewLine;
```

### 1.3. Приём сообщения системой

Приём сообщения осуществляется посредством создания потоков. Чтобы прослушивать заданный порт непрерывно, а не по нажатию кнопки, создаётся отдельный поток, который этим занимается, периодически «засыпающий», чтобы постоянно не грузить систему. На каждый прослушиваемый порт создаётся свой поток, благодаря чему можно принимать сообщение одновременно со всех портов. Если в поток приёма данных попадает байтовый массив, переданный из одного окна другому, то этот массив переводится в строковую переменную.

### 1.4. Передача сообщений между окнами

В окнах злоумышленника и компьютеров сети предусмотрена возможность отправить сообщение на выбранный адрес. При этом если отправитель не знает входящий порт адресата (имитация MAC-адреса), то будет послан широковещательный пакет ARP-Request. Также был добавлен обработчик, проверяющий, что за сообщение нам пришло, и либо отбрасывающий его, либо инициализирующий какие-то определённые действия.

Например, если пришёл ARP-Request, то в ответ отправляется ARP-Reply:

```
//Если ARP-Request
if (instr.Substring(13, 1) == "1")
{
    textBox.Text += ««ARP REQUEST от " + instr.Substring(10, 3);
    textBox.Text += Environment.NewLine;
    //Обычное + Mac-адрес назначения + наш Mac-адрес +
    адрес назначения + наш адрес + ARP-Reply
```

```
str += "0" + instr.Substring(4, 3) + localPort + instr.Substring(10, 3) +  
adres + "2";  
Thread.Sleep(1000);  
                //Отправка  
Send(str);  
return;  
}
```

### **1.5. Окно сервера**

Создано окно сервера. Оно принимает не ARP сообщения только после процедуры «тройного рукопожатия».

После получения пакета SYN сервер записывает исходящий MAC-адрес из пакета в буфер и отправляет ответ SYN-ACK. Если затем придёт пакет ACK от той же машины, то сервер «запомнит» подключение и уберёт соответствующую запись из буфера.

Чистку буфера через некоторое время, как на настоящем оборудовании, решено не добавлять, так как программа работает значительно медленнее реальных сетей, и для демонстрации это не понадобится.

### **1.6. Конфигурирование сети**

В программе конфигурирование сети производится посредством кнопок «Добавить маршрутизатор», «Добавить коммутатор», «Добавить сервер», «Добавить компьютер».

### **1.7. Окно коммутатора**

Создано окно коммутатора. Принимая на один из портов пакет, коммутатор сначала проверяет MAC-адрес отправителя и ищет его в своей таблице MAC-адресов.

Если не находит, то делает в ней запись, что этот MAC доступен по порту, с которого пришло сообщение. Затем проверяется MAC-адрес назначения и также ищется в таблице. Если находится, то пакет посылается на нужный порт, если не находится — происходит широковещательная рассылка.

Для того чтобы программа нормально функционировала, на все «устройства» добавлены переменные или их массивы (в зависимости от типа), хранящие информацию о том, какие «соседские» порты к каким локальным портам подключены (имитация соединения проводом). Иначе окна знали бы только на какой порт пришёл пакет, но не с какого на соседнем «устройстве».

### **1.8. Окно маршрутизатора**

Создано окно маршрутизатора. Его отличает от коммутатора то, что он работает с обычными адресами, а не с MAC-адресами. А также то, что маршрутизаторы обмениваются между собой таблицами маршрутизации.

В программе это происходит при изменении топологии сети, инициализированном с одного из них.

### 1.9. Тестирование программы

Программа написана на языке С#.

Программа должна имитировать разные типы атак на компьютерную сеть. Атаки тестируются. Если что-то будет работать ненадлежащим образом, то следует найти причины ошибки и исправить её.

## 2. Запуск программы и работа с ней

### 2.1. Запуск программы

При запуске программы появляется стартовое окно — окно злоумышленника (рис. 1).

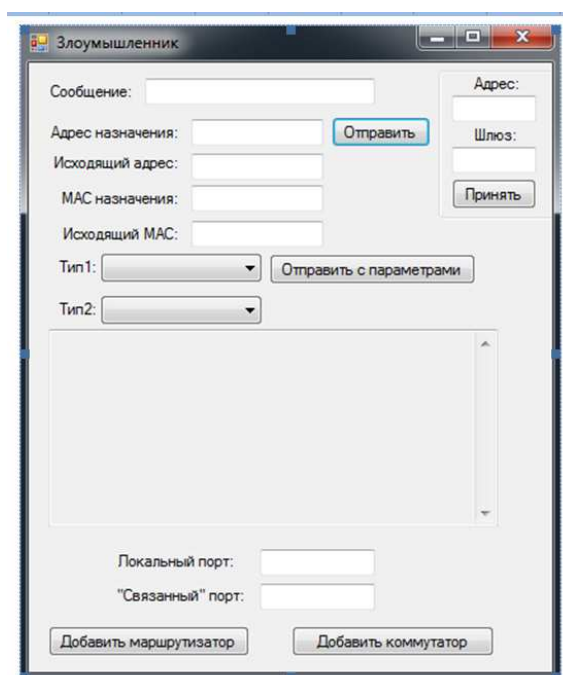


Рис. 1. Окно злоумышленника

Мы видим, что злоумышленник имеет возможность отправить произвольное сообщение на указанный адрес внутри моделируемой сети, подключить новый маршрутизатор или коммутатор.

Имеется поле, куда будут выводиться полученные и служебные сообщения.

Основное отличие этого окна от других «хостов» — это наличие кнопки **«Отправить с параметрами»**, позволяющей, используя введенные пользователем данные, передать сообщение с произвольными адресами назначения и отправки, а также выбранного типа.

## 2.2. Конфигурирование сети

Чтобы создать компьютерную сеть, которую злоумышленнику предстоит атаковать, нужно из окна злоумышленника «найти», породить маршрутизатор или коммутатор, принадлежащий этой сети.

Это новое устройство будет сразу подключено к компьютеру злоумышленника «проводом». Для это следует в два поля ввести **трёхзначные адреса**, первый в поле «Локальный порт» — это порт компьютера злоумышленника, с которого идёт подключение, второй — в поле «Связанный порт» — подключённого «проводом» порта второго создаваемого устройства, затем нажать «Добавить маршрутизатор» или «Добавить коммутатор».

## 2.3. Порождение хостов сети

В открытом окне порождённого маршрутизатора или коммутатора можно добавлять в сеть новый компьютер. Добавленная машина сразу соединится с родительским устройством (с тем, с которого она была открыта).

В созданном хосте нет возможности добавлять новые узлы в сеть, так как в данной программе он является одной из конечных точек сети.

## 2.4. Добавление сервера

Добавление сервера производится из окна коммутатора. Его отличие от компьютера в том, что он не принимает входящие сообщения, пока не будет установлено соединение процессом «тройного рукопожатия».

# 3. Имитация атак

Программа активно использует многопоточность для симуляции сети, в особенности это важно для коммутаторов и маршрутизаторов, т. к. им приходится прослушивать сразу несколько портов. Сообщения между окнами передаются через TCP-клиенты потоковой передачей.

Программа позволяет имитировать атаки типа Smurfing (ICMP-Flood), SYN-Flood, подмена MAC-адреса, ARP-spoofing.

## 3.1. Smurfing (ICMP-Flood)

Для демонстрации взрывного увеличения сетевого трафика в окне злоумышленника имеется возможность отослать пакет ICMP ECHO REQUEST (поле «Тип 1»). В окнах компьютеров, в свою очередь, добавлена функция обработки этого условного пакета и отправки в ответ ICMP ECHO REPLY.

Для отправки такого сообщения с компьютера 200, принадлежащего злоумышленнику, с помощью кнопки «Отправить с параметрами» надо выбрать данные, как на рисунке (см. рис. 2, слева). Тогда после первой отправки инициализируется ARP знакомство хостов, а после второй уже пойдут пакеты ICMP ECHO REPLY на машину жертвы (компьютер 300, см. рис. 2, справа). Как

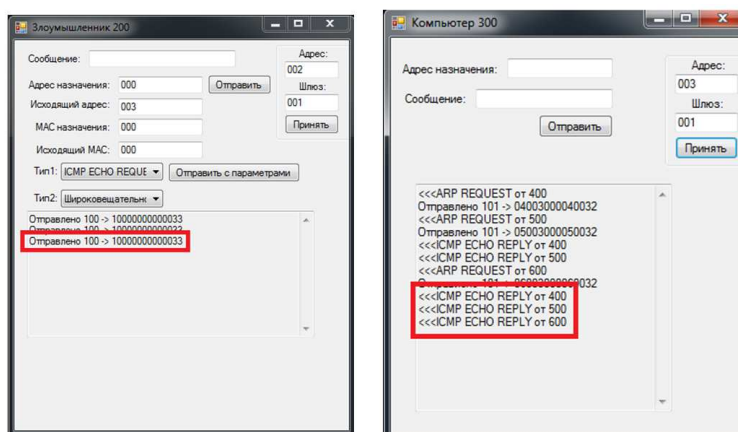


Рис. 2. Атака Smurfing. Окно злоумышленника слева, окно жертвы – справа

видно из рисунка, количество сообщений на канале передачи данных жертвы равно количеству хостов в сети (не считая двух «главных героев») при всего лишь одном сообщении у злоумышленника.

Цикл широковещательной отправки:

```
//отправка по всем каналам, кроме того, откуда пришло
for (int i = 0; i < llen; i++)
{
    if (localPorts[i] != port)
    {
        Thread.Sleep(200);
        Send(outPorts[i], str);
    }
}
```

### 3.2. SYN-Flood

SYN-Flood использует механизм установления TCP-соединения. Есть три состояния: посылка SYN-пакета, получение пакета SYN-ACK и посылка ACK-пакета.

Идея атаки состоит в создании большого количества не до конца установленных (полуоткрытых) TCP-соединений. При этом злоумышленник создаёт и от имени любых, даже никому не принадлежащих, IP-адресов направляет на атакуемый сервер большое количество запросов на установление соединения (пакеты с флагом SYN). Атакуемая машина отвечает пакетами подтверждениями SYN-ACK на ложные адреса отправителей в интернете, которые в результате или не найдут адресата, или «озадачат» операционную систему на действующем IP-адресе (поскольку никаких запросов эта система на данный сервер не отправляла) и будут проигнорированы.

Для каждого полученного SYN-пакета сервер-жертва выделяет место в буфере, в результате исчерпания ресурсов которого новые соединения (даже настоящие) не могут быть открыты, поскольку протокол сеансового уровня не сможет вычислить фальшивые запросы с тем, чтобы установить больший приоритет настоящим.

Для начала демонстрации существенных задержек при установке связи нужно добавить в сеть сервер.

Его отличие от компьютера в том, что он не принимает входящие сообщения, пока не будет установлено соединение процессом «тройного рукопожатия».

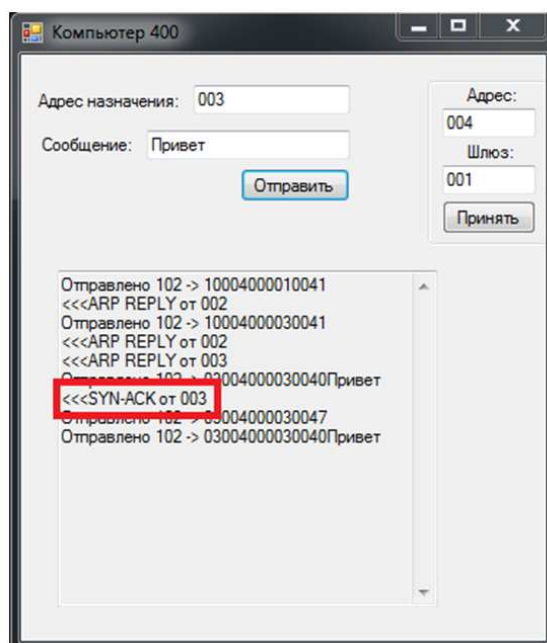


Рис. 3. Ответ SYN-ACK от сервера

В обычной ситуации сервер отвечает пакетом SYN-ACK на все новые подключения (см. рис. 3). Но если злоумышленник начнёт непрерывно посылать от чужих адресов пакеты SYN на сервер, последний будет добавлять эти запросы в очередь. А поскольку злоумышленник отвечать на SYN-ACK сервера не будет, очередь переполнится, и не останется места новым подключениям. Даже при условии чистки очереди через некоторое время (тайм-аут подключений) задержки будут серьёзными.

### 3.3. Подмена MAC-адреса

Хакер может легко ассоциировать рабочий IP-адрес с фальшивым MAC-адресом. Например, он посылает ARP-ответ, который связывает IP вашего роутера с несуществующим MAC-адресом. Ваши компьютеры уверены в том, что они знают шлюз по умолчанию, но в реальности они шлют пакеты неизвестно

куда и неизвестно кому. Одним легким движением злоумышленник отрезал вас от интернета.

Для демонстрации изменения MAC-адреса в ARP-таблице для начала отправим сообщение «Первый привет» с машины жертвы с внешним адресом назначения (в программе это 100 и больше) на заранее созданный компьютер, находящийся «за» маршрутизатором и имитирующим интернет. Послание дойдёт до адресата.

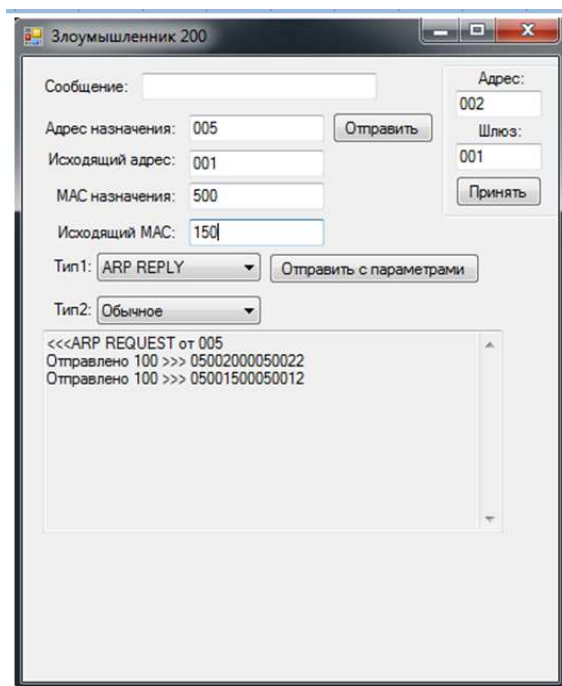


Рис. 4. Отправка поддельного ARP-Reply от хакера

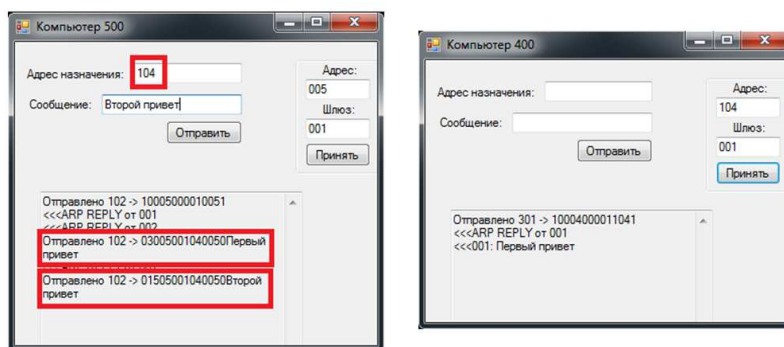


Рис. 5. Отправка двух сообщений с машины (слева). Получение первого сообщения (справа)

Затем отправим от злоумышленника ARP-Reply с нужными параметрами (см. рис. 4), чтобы машина жертвы, сопоставив IP из пакета с IP роутера,



заменяла его MAC-адрес на новый. После этого попытка снова отправить сообщение «Второй привет» (см.рис. 5, слева) не увенчается успехом (см. рис. 5, справа). Теперь жертва отрезана от интернета.

### 3.4. ARP-spoofing

Программа демонстрирует возможности перехвата сетевого трафика. Злоумышленник посылает поддельный пакет ARP reply роутеру, который, сопоставив IP из пакета с IP жертвы, заменяет её MAC-адрес на MAC-адрес злоумышленника. Затем посылается поддельный пакет ARP reply жертве, которая, сопоставив IP из пакета с IP роутера, заменяет его MAC-адрес на MAC-адрес злоумышленника.

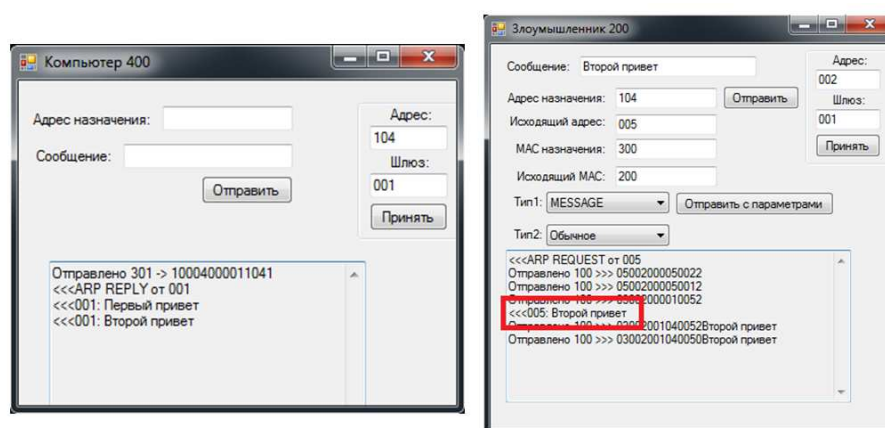


Рис. 6. ARP-spoofing. Окно злоумышленника слева, окно жертвы – справа.

Теперь злоумышленник может просматривать исходящий сетевой трафик жертвы. Данная атака весьма похожа на подмену MAC-адреса, отличие в том, что второе сообщение всё-таки доходит до компьютера-адресата (см. рис. 6, слева), но не напрямую, а предварительно пройдя через машину злоумышленника (см. рис. 6, справа).

## ЛИТЕРАТУРА

1. Степашкин М.В., Котенко И.В., Богданов В.С. Моделирование атак для активного анализа уязвимостей компьютерных сетей // Вторая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» ИММОД–2005. С. 269–273.
2. Шоров А.В. Имитационное моделирование механизмов защиты компьютерных сетей от инфраструктурных атак на основе подхода «нервная система сети». Автореферат диссертации на соискание учёной степени кандидата технических наук. Санкт-Петербург, 2012.

3. Тумоян Е.П., Кавчук Д.А. Моделирование сетевых атак в задачах автоматического анализа защищённости информационных систем // Системы обработки информации. 2012. Т. 2, вып.4. С. 74–78.
4. Мордвин Д.В., Абрамов Е.С. Разработка инструментальных средств для моделирования ЛВС // Известия ЮФУ. Технические науки. 2007. № 4. С. 113–116.
5. Эннс Е.П. Уязвимости сетевых протоколов и их моделирование: Выпускная квалификационная работа. ОмГУ, кафедра кибернетики, 2017. 29 с.

## PROGRAM FOR SIMULATION OF COMPUTER NETWORK AND NETWORK ATTACKS

**A.K. Guts**

Dr.Sc. (Phys.-Math.), Professor, e-mail: guts@omsu.ru

**E.P. Enns**

Student, e-mail: jecka-gtx@mail.ru

Dostoevsky Omsk State University

**Abstract.** We present a computer program that simulates the work of computer network and allows us to demonstrate the result of malicious attacks on network using several existing network vulnerability.

**Keywords:** computer network, network attacks, attack simulation, computer program.

*Дата поступления в редакцию: 02.07.2017*

## **ИДЕНТИФИКАЦИЯ СУИЦИДАЛЬНЫХ ГРУПП И НАРУШИТЕЛЕЙ АВТОРСКИХ ПРАВ В СОЦИАЛЬНЫХ СЕТЯХ**

**А.В. Костылев**

студент, e-mail: kostart\_94@mail.ru

**Д.Н. Лавров**

к.т.н., доцент, e-mail: dmitry.lavrov72@gmail.com

**А.К. Гуц**

д.ф.-м.н., профессор, e-mail: aguts@mail.ru

Омский государственный университет им. Ф.М. Достоевского

**Аннотация.** В данной работе поставлена задача разработки программного обеспечения, позволяющего автоматизировать поиск групп и пользователей в социальных сетях, размещающих в сеть суицидальные материалы и материалы, нарушающие авторское право. В качестве социальной сети для исследования возможности подобного поиска была выбрана социальная сеть «ВКонтакте» — крупнейшая социальная сеть СНГ. В основу разрабатываемого программного обеспечения был положен алгоритм латентного семантического анализа. В статье обсуждаются основные варианты алгоритма и подобран наиболее адекватно работающий на реальных данных вариант.

**Ключевые слова:** классификация текстов, латентно-семантический анализ, авторское право, правонарушения, социальные сети, суицидальные группы.

### **Введение**

Социальные сети стремительно входят в современную социальную реальность. По состоянию на декабрь 2016 года количество пользователей сети «Facebook» превысило 1700 миллионов человек, а сети «ВКонтакте» — 400 миллионов. Согласно опросу исследовательского холдинга «Ромир» более 90% российских пользователей интернета ведут активную социальную жизнь в сети [1].

Из-за стремительного увеличения популярности социальных сетей они стали источником правонарушений. По данным информационно-аналитического центра «Сова», только в 2015 году за посты и репосты в социальной сети «ВКонтакте» к уголовной ответственности привлекли 119 человек [2].

При этом отсутствуют какие-либо данные об автоматизации поиска правонарушений: до сих пор большинство проверок выполняется по жалобам других

пользователей, которые могут быть не всегда заинтересованы в выявлении нарушения. Во многом отсутствие подобных методов можно объяснить новизной способов правонарушений, их непохожестью на прочие способы и относительно недавним осознанием необходимости борьбы с ними.

В данной работе поставлена задача разработки программного обеспечения, позволяющего автоматизировать поиск групп и пользователей в социальных сетях, размещающих суицидальные материалы и материалы, нарушающие авторское право. В дальнейшем по тексту статьи мы будем называть такие материалы запрещёнными.

## **1. Выбор инструментов и подготовка к созданию приложения**

### **1.1. Выбор инструментов разработки**

В качестве социальной сети для исследования возможности подобного поиска была выбрана социальная сеть «ВКонтакте» — крупнейшая социальная сеть СНГ. Как уже было сказано, по состоянию на декабрь 2016 года в ней имеется более 400 миллионов уникальных пользователей, в её базе данных учётных записей представлена огромная доля российского населения. Данная сеть имеет собственное API — VK API, позволяющее осуществлять самые разнообразные запросы.

В качестве языка разработки было принято решение использовать язык программирования Java. Так как в прошлом значительная часть исходного кода уже была написана на языке Python, чтобы избежать его переписывания, был использован Jython — реализация языка Python на языке Java. В качестве сборщика проекта использовался фреймворк Maven. Для обращения к VK API и разбора полученных результатов использовались библиотеки Apache HttpClient и Gson. Графический интерфейс был написан с использованием Swing. Различные полезные функции линейной алгебры были взяты из библиотеки Jama. Для логирования использовались Slf4J и Log4j.

Для взаимодействия с VK API можно использовать библиотеку: vk-java-sdk (<https://github.com/VKCOM/vk-java-sdk>). Однако для достижения большей гибкости запросов и для гарантированной корректной работы API-методов было решено создать свой API, а не пользоваться готовой реализацией.

### **1.2. Создание standalone-приложения**

Все методы VK API можно подразделить на 2 типа: открытые, не требующие ключа доступа, и закрытые, для которых он необходим. К закрытым методам относятся, в частности, методы для работы с аудио- и видеоматериалами.

Для того, чтобы получить возможность обращаться ко всем методам VK API, необходимо зарегистрировать своё standalone-приложение.

Процесс регистрации приложения в социальной сети «ВКонтакте» весьма прост: достаточно указать имя приложения и его тип — «standalone» (рис. 1).

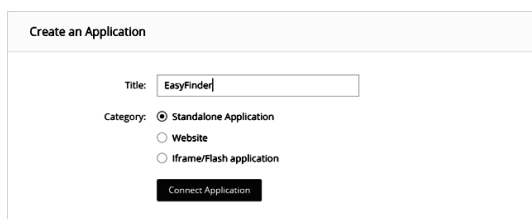


Рис. 1. Создание standalone-приложения

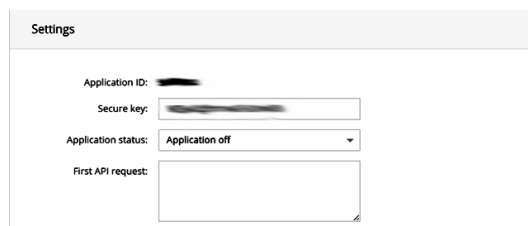


Рис. 2. Полученные Application Id и Secure key

При регистрации приложения в сети «ВКонтакте» ему присваиваются два параметра — Application ID и Secure key (рис. 2). В дальнейшем они будут необходимы при авторизации пользователя «ВКонтакте», от имени которого и будут выполняться запросы к закрытым методам API.

После выполнения этих шагов можно переходить к разработке самого приложения.

### 1.3. Авторизация

Для того чтобы получить доступ ко всем возможностям VK API, необходимо выполнить вход с одного из аккаунтов социальной сети и дать приложению необходимые права доступа. Для выполнения входа требуется отправить get-запрос по адресу <http://oauth.vk.com/oauth/authorize> с параметрами:

- `client_id` = id\_нашего\_приложения
- `scope` = 'offline, video, audio' — для получения бессрочного доступа к API видео- и аудиофайлам
- `response_type` = 'token'
- `v` = 5.52 — актуальная версия VK API

В случае удачного выполнения запроса вернётся значение `access_token`, необходимое для осуществления запросов к закрытым методам VK API.

### 1.4. Ограничения VK API

В ходе исследования документации VK API выяснилось, что оно имеет несколько ограничений:

- максимальное количество возвращаемых объектов: 100 для `wall.get`, 200 для `video.get`;
- максимальное количество запросов — 3 запроса в секунду.

Таким образом, при создании функций использовалась следующая логика: API-запросы отправлялись со смещением (offset), каждую треть секунды до тех пор, пока количество полученных очередным API-запросом результатов не становилось равным нулю.

Недостатком данного подхода является относительно длительное время выполнения запроса. Данный недостаток можно устранить, используя API-метод `execute`, позволяющий выполнить произвольный код на языке VKScript

— аналоге языка JavaScript, по уверениям разработчиков, совместимом с ECMAScript. В предыдущей работе было показано, что, несмотря на некоторую ограниченность возможностей данного подхода, с его помощью можно добиться увеличения скорости выполнения запросов к VK API до 6,5 раз.

Также необходимо упомянуть, что 16 декабря 2016 года руководство «ВКонтакте» отключило доступ к публичному музыкальному API сторонним приложениям [3].

## **2. Разработка части приложения, определяющей предпочтения пользователей и тематику групп по размещённым постам**

При решении задачи поиска групп и пользователей социальной сети «ВКонтакте», размещающих в сеть запрещённые материалы, была рассмотрена более общая задача: выявление предпочитаемых тем постов пользователя и определение тематики групп. Решив данную задачу, несложно искать только пользователей и группы с определёнными «интересами». Было принято решение воспользоваться методом латентно-семантического анализа, предназначенного для выявления скрытых связей между документами, о котором будет рассказано ниже.

### **2.1. Получение постов пользователя/группы**

Для того чтобы проанализировать записи пользователя или группы в сети «ВКонтакте», их необходимо получить с помощью VK API. Для этого требуется вызвать только один API-метод: `wall.get`, возвращающий записи пользователя/группы на стене. Данный метод может вернуть до 100 результатов, чтобы получить все результаты, его нужно вызвать несколько раз. Для обращения к VK API была использована библиотека Apache HttpClient, для парсинга получившихся JSON-данных использована библиотека Gson. На основе вышеперечисленных технологий созданы функции `Set<Post> getPostsByUserId(Integer owner_id, boolean isOwnerOnly)` и `Set<Post> getPostsByGroupId(Integer group_id)`, возвращающие список постов на стене пользователя/группы. Переменная `isOwnerOnly` в случае пользователя определяет, возвращать все посты или только посты непосредственно данного пользователя.

### **2.2. Обработка полученных данных, латентно-семантический анализ**

Из получившихся на прошлом этапе постов выделяется текст, из него отфильтровываются небуквенные символы, остальные символы переводятся в нижний регистр. Из полученных строк строится список слов, присутствующих в постах, а из него составляется матрица  $A$  наличия слов в постах. Строки матрицы — все слова, встречающиеся хотя бы в одном из постов, столбцы —

номера постов пользователя/группы, а в пересечениях стоят либо единицы, если данное слово есть в данном посте, либо нули, если нет (рис. 3).

Для классификации текстов может использоваться методы на основе  $k$ -средних, хорошо зарекомендовавший себя для классификации научных текстов [7]. Тем не менее, было принято решение применить метод латентно-семантического анализа, который основывается на сингулярном разложении матрицы  $A$  с понижением ранга: матрица  $A$  раскладывается на 3 матрицы:  $U$ ,  $\Sigma$ ,  $V^T$ , где матрицы  $U$  и  $V$  — это две унитарные матрицы, состоящие из левых и правых сингулярных векторов соответственно, а  $\Sigma$  — диагональная матрица, значения на диагонали которой называются сингулярными значениями (числами) матрицы  $A$ .

Особенность сингулярного разложения заключается в том, что если в матрице  $\Sigma$  оставить только  $k$  наибольших сингулярных значений, то комбинация полученных матриц  $\hat{A} = U \cdot \Sigma \cdot V^T$  будет наилучшим приближением исходной матрицы  $A$  к матрице ранга  $k$ .

Ранг полученной матрицы требуется понизить из-за того, что исходная матрица содержит множество «шумов». Однако чрезмерное понижение ранга матрицы ведёт к потере значимой информации. Существует множество мнений по поводу выбора наилучшего значения ранга матрицы  $k$ : например, в работе А.Д. Хомоненко и С.А. Краснова показывается, что для решения задачи по разделению двух различных групп наилучшим рангом является  $k = 2$  [4].

Получившуюся матрицу  $\hat{A}$  можно рассмотреть как матрицу вектор-значений, соответствующих постам  $T_i$  в  $n$ -размерном пространстве. Из этой матрицы можно составить матрицу корреляции векторов  $P$ , — например, подсчётом косинуса угла для каждой пары вектор-значений из матрицы  $A$ .

Также, согласно данным исследований [5, 6], получившуюся матрицу  $V^T$  можно рассмотреть в качестве как вектор-значений, соответствующих постам  $T_i$  в  $n$ -размерном пространстве. Если же взять  $k = 2$ , то каждому посту  $T_i$  будет соответствовать пара значений  $(x, y)$ , которые очень удобно представить в виде графика.

Существует множество готовых библиотек для Java, позволяющих использовать сингулярное разложение: Jama, ujmp, Commons Math, colt, jblas. В нашей программе была использована Jama (Java Matrix Library) — библиотека функций линейной алгебры, созданная в NIST и являющаяся общественным достоянием.

### 2.3. Проверка работы латентно-семантического анализа на тестовых данных

Для проверки корректности и демонстрации работы алгоритма сингулярного разложения было написано две функции: первая рассчитывала матрицу

|            | $T_1$ | $T_2$ | $T_3$ |
|------------|-------|-------|-------|
| бензол     | 1     | 1     | 0     |
| углевод    | 1     | 0     | 1     |
| бензин     | 1     | 0     | 1     |
| пластмасса | 0     | 1     | 0     |
| синтез     | 0     | 0     | 1     |

Рис. 3. Пример внешнего вида матрицы  $A$

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,98 | 0,97 | 0,98 | 1    | 0,98 | 0,34 | 0,37 | 0,31 | 0,24 | 0,41 | 0,31 | 0,59 | 0,63 | 0,65 | 0,72 | 0,65 | 0,6  |
| 0,98 | 1    | 1    | 1    | 0,99 | 1    | 0,17 | 0,21 | 0,14 | 0,07 | 0,25 | 0,14 | 0,44 | 0,48 | 0,5  | 0,59 | 0,5  | 0,45 |
| 0,97 | 1    | 1    | 1    | 0,97 | 1    | 0,09 | 0,12 | 0,05 | 0    | 0,16 | 0,05 | 0,36 | 0,41 | 0,42 | 0,52 | 0,42 | 0,37 |
| 0,98 | 1    | 1    | 1    | 0,98 | 1    | 0,14 | 0,17 | 0,1  | 0,03 | 0,21 | 0,1  | 0,41 | 0,45 | 0,47 | 0,56 | 0,47 | 0,41 |
| 1    | 0,99 | 0,97 | 0,98 | 1    | 0,98 | 0,32 | 0,35 | 0,29 | 0,22 | 0,39 | 0,29 | 0,57 | 0,61 | 0,63 | 0,7  | 0,63 | 0,57 |
| 0,98 | 1    | 1    | 1    | 0,98 | 1    | 0,15 | 0,18 | 0,11 | 0,04 | 0,22 | 0,11 | 0,42 | 0,46 | 0,48 | 0,57 | 0,48 | 0,42 |
| 0,34 | 0,17 | 0,09 | 0,14 | 0,32 | 0,15 | 1    | 1    | 1    | 0,99 | 1    | 1    | 0,96 | 0,95 | 0,94 | 0,9  | 0,94 | 0,96 |
| 0,37 | 0,21 | 0,12 | 0,17 | 0,35 | 0,18 | 1    | 1    | 1    | 0,99 | 1    | 1    | 0,97 | 0,96 | 0,95 | 0,91 | 0,95 | 0,97 |
| 0,31 | 0,14 | 0,05 | 0,1  | 0,29 | 0,11 | 1    | 1    | 1    | 1    | 0,99 | 1    | 0,95 | 0,93 | 0,93 | 0,88 | 0,93 | 0,95 |
| 0,24 | 0,07 | 0    | 0,03 | 0,22 | 0,04 | 0,99 | 0,99 | 1    | 1    | 0,98 | 1    | 0,92 | 0,91 | 0,9  | 0,85 | 0,9  | 0,92 |
| 0,41 | 0,25 | 0,16 | 0,21 | 0,39 | 0,22 | 1    | 1    | 0,99 | 0,98 | 1    | 0,99 | 0,98 | 0,97 | 0,96 | 0,93 | 0,96 | 0,98 |
| 0,31 | 0,14 | 0,05 | 0,1  | 0,29 | 0,11 | 1    | 1    | 1    | 1    | 0,99 | 1    | 0,95 |      |      | 0,88 | 0,93 | 0,95 |
| 0,59 | 0,44 | 0,36 | 0,41 | 0,57 | 0,42 | 0,96 | 0,97 | 0,95 | 0,92 | 0,92 | 0,95 | 1    | 1    | 1    | 0,99 | 1    | 1    |
| 0,63 | 0,48 | 0,41 | 0,45 | 0,61 | 0,46 | 0,95 | 0,96 | 0,93 | 0,91 | 0,97 | 0,93 | 1    | 1    | 1    | 0,99 | 1    | 1    |
| 0,65 | 0,5  | 0,42 | 0,47 | 0,63 | 0,48 | 0,94 | 0,95 | 0,93 | 0,9  | 0,96 | 0,93 | 1    | 1    | 1    | 0,99 | 1    | 1    |
| 0,72 | 0,59 | 0,52 | 0,56 | 0,7  | 0,57 | 0,9  | 0,91 | 0,88 | 0,85 | 0,93 | 0,88 | 0,99 | 0,99 | 0,99 | 1    | 0,99 | 0,99 |
| 0,65 | 0,5  | 0,42 | 0,47 | 0,63 | 0,48 | 0,94 | 0,95 | 0,93 | 0,9  | 0,96 | 0,93 | 1    | 1    | 1    | 0,99 | 1    | 1    |
| 0,6  | 0,45 | 0,37 | 0,41 | 0,57 | 0,42 | 0,96 | 0,97 | 0,95 | 0,92 | 0,98 | 0,95 | 1    | 1    | 1    | 0,99 | 1    | 1    |

Рис. 5. Матрица корреляций  $P$ ,  $k = 2$ . Чёрным цветом отмечены блоки постов на определённую тему, красным — неожиданно высокая корреляция между постами второй и третьей групп

$\hat{A} = U \cdot \Sigma \cdot V^T$  с заданным рангом  $k$  и выводила её на экран, вторая ставила в соответствие списку постов пользователя список пар  $(x, y)$  и выводила результаты в графическом виде.

Далее выбрана одна из страниц сети «ВКонтакте», на ней было создано 18 постов со следующей тематикой:

- 6 постов на тему кулинарии;
- 6 постов на тему органической химии;
- 6 постов на тему футбола.

Результат работы функции, ставящей в соответствие списку постов пользователя список пар  $(x, y)$ , представлен на рисунке 4:

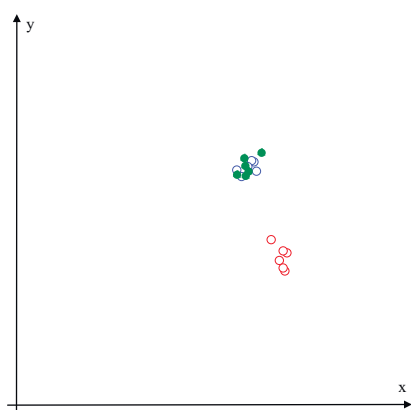


Рис. 4. Вектор-значения матрицы  $V^T$ , соответствующие постам. Значения по обеим осям изменяются в промежутке от 0 до 1. Красные круги — посты на тему футбола, зелёные — химии, синие — кулинарии

Из приведённого рисунка можно заметить, что посты на тему футбола формируют отдельную группу, в то время как посты на тему кулинарии и химии сильно слились.

На рисунке 5 приведены эксперименты по определению группы предпочтений по значениям матрицы корреляций.

Можно заметить, что в матрице корреляции наблюдается схожая картина — посты на тему кулинарии и химии сильно слились. Как оказалось, это было вызвано малым выбранным рангом  $k$ . Ниже приведены матрицы корреляций для  $k = 3$  и  $k = 4$ .

Из рисунков 6 и 7 видно, что при



|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,97 | 0,89 | 0,73 | 1    | 0,97 | 0,36 | 0,38 | 0,33 | 0,26 | 0,43 | 0,33 | 0,4  | 0,31 | 0,42 | 0,44 | 0,37 | 0,41 |
| 0,97 | 1    | 0,97 | 0,63 | 0,97 | 1    | 0,26 | 0,24 | 0,19 | 0,17 | 0,29 | 0,21 | 0,17 | 0,08 | 0,19 | 0,22 | 0,15 | 0,18 |
| 0,89 | 0,97 | 1    | 0,43 | 0,87 | 0,97 | 0,25 | 0,19 | 0,15 | 0,19 | 0,24 | 0,19 | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,73 | 0,63 | 0,43 | 1    | 0,77 | 0,63 | 0    | 0,03 | 0    | 0    | 0,06 | 0    | 0,61 | 0,67 | 0,66 | 0,72 | 0,67 | 0,61 |
| 1    | 0,97 | 0,87 | 0,77 | 1    | 0,96 | 0,32 | 0,35 | 0,29 | 0,22 | 0,39 | 0,29 | 0,42 | 0,34 | 0,44 | 0,47 | 0,4  | 0,42 |
| 0,97 | 1    | 0,97 | 0,63 | 0,96 | 1    | 0,24 | 0,22 | 0,17 | 0,15 | 0,27 | 0,19 | 0,15 | 0,06 | 0,17 | 0,2  | 0,12 | 0,15 |
| 0,36 | 0,26 | 0,25 | 0    | 0,32 | 0,24 | 1    | 0,98 | 0,99 | 0,99 | 0,98 | 1    | 0,49 | 0,26 | 0,43 | 0,36 | 0,36 | 0,5  |
| 0,35 | 0,24 | 0,19 | 0,03 | 0,35 | 0,22 | 0,98 | 1    | 1    | 0,96 | 1    | 0,99 | 0,65 | 0,46 | 0,6  | 0,53 | 0,53 | 0,65 |
| 0,33 | 0,19 | 0,15 | 0    | 0,29 | 0,17 | 0,99 | 1    | 1    | 0,98 | 0,99 | 1    | 0,61 | 0,42 | 0,56 | 0,48 | 0,49 | 0,62 |
| 0,26 | 0,17 | 0,19 | 0    | 0,22 | 0,15 | 0,99 | 0,96 | 0,98 | 1    | 0,96 | 0,99 | 0,43 | 0,22 | 0,37 | 0,29 | 0,3  | 0,44 |
| 0,43 | 0,29 | 0,24 | 0,06 | 0,39 | 0,27 | 0,98 | 1    | 0,99 | 0,96 | 1    | 0,99 | 0,65 | 0,45 | 0,6  | 0,53 | 0,53 | 0,65 |
| 0,33 | 0,21 | 0,19 | 0    | 0,29 | 0,19 | 1    | 0,99 | 1    | 0,99 | 0,99 | 1    | 0,56 | 0,35 | 0,5  | 0,42 | 0,43 | 0,56 |
| 0,4  | 0,17 | 0    | 0,61 | 0,42 | 0,15 | 0,49 | 0,65 | 0,61 | 0,43 | 0,65 | 0,56 | 1    | 0,97 | 1    | 0,99 | 0,99 | 1    |
| 0,31 | 0,08 | 0    | 0,67 | 0,34 | 0,06 | 0,28 | 0,46 | 0,42 | 0,22 | 0,45 | 0,35 | 0,97 | 1    | 0,98 | 0,99 | 1    | 0,97 |
| 0,42 | 0,19 | 0    | 0,66 | 0,44 | 0,17 | 0,43 | 3,6  | 0,56 | 0,37 | 0,6  | 0,5  | 1    | 0,98 | 1    | 1    | 1    | 1    |
| 0,44 | 0,22 | 0    | 0,72 | 0,47 | 0,2  | 0,36 | 0,53 | 0,48 | 0,29 | 0,53 | 0,42 | 0,99 | 0,99 | 1    | 1    | 1    | 0,99 |
| 0,37 | 0,15 | 0    | 0,67 | 0,4  | 0,12 | 0,36 | 0,53 | 0,49 | 0,3  | 0,53 | 0,43 | 0,99 | 1    | 1    | 1    | 1    | 0,99 |
| 0,41 | 0,18 | 0    | 0,61 | 0,42 | 0,15 | 0,5  | 0,65 | 0,62 | 0,44 | 0,65 | 0,56 | 1    | 0,97 | 1    | 0,99 | 0,99 | 1    |

Рис. 6. Матрица корреляций  $P$ ,  $k = 3$ 

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,81 | 0,8  | 0,19 | 0,97 |      | 0,07 | 0,16 | 0,17 | 0    | 0,2  | 0,11 | 0,59 | 0,52 | 0,53 | 0,6  | 0,55 | 0,58 |
| 0,81 | 1    | 0,96 | 0,52 | 0,9  | 0,99 | 0,24 | 0,23 | 0,19 | 0,16 | 0,26 | 0,2  | 0,15 | 0,07 | 0,18 | 0,19 | 0,13 | 0,16 |
| 0,8  | 0,96 | 1    | 0,26 | 0,85 | 0,97 | 0,19 | 0,15 | 0,13 | 0,13 | 0,2  | 0,15 | 0,02 | 0    | 0,01 | 0,04 | 0    | 0,02 |
| 0,19 | 0,52 | 0,28 | 1    | 0,39 | 0,45 | 0,13 | 0,18 | 0,06 | 0,06 | 0,19 | 0,09 | 0,11 | 0,16 | 0,29 | 0,22 | 0,19 | 0,14 |
| 0,97 | 0,9  | 0,85 | 0,39 | 1    | 0,93 | 0,14 | 0,22 | 0,21 | 0,05 | 0,27 | 0,16 | 0,53 | 0,46 | 0,52 | 0,56 | 0,5  | 0,52 |
| 0,86 | 0,99 | 0,97 | 0,45 | 0,93 | 1    | 0,18 | 0,18 | 0,15 | 0,1  | 0,23 | 0,15 | 0,18 | 0,11 | 0,2  | 0,23 | 0,16 | 0,19 |
| 0,07 | 0,24 | 0,19 | 0,14 | 0,14 | 0,16 | 1    | 0,98 | 0,96 | 1    | 0,98 | 0,99 | 0,18 | 0,01 | 0,23 | 0,09 | 0,1  | 0,21 |
| 0,16 | 0,23 | 0,15 | 0,16 | 0,22 | 0,16 | 0,98 | 1    | 0,99 | 0,96 | 1    | 0,99 | 0,37 | 0,22 | 0,43 | 0,3  | 0,3  | 0,4  |
| 0,17 | 0,19 | 0,13 | 0,06 | 0,21 | 0,15 | 0,96 | 0,99 | 1    | 0,95 | 0,99 | 0,99 | 0,42 | 0,26 | 0,45 | 0,32 | 0,33 | 0,44 |
| 0    | 0,16 | 0,13 | 0,06 | 0,05 | 0,1  | 1    | 0,96 | 0,95 | 1    | 0,96 | 0,98 | 0,12 | 0    | 0,17 | 0,03 | 0,04 | 0,15 |
| 0,2  | 0,28 | 0,2  | 0,19 | 0,27 | 0,23 | 0,98 | 1    | 0,99 | 0,96 | 1    | 0,99 | 0,38 | 0,23 | 0,44 | 0,31 | 0,31 | 0,41 |
| 0,11 | 0,2  | 0,15 | 0,09 | 0,16 | 0,15 | 0,99 | 0,99 | 0,99 | 0,98 | 0,99 | 1    | 0,3  | 0,14 | 0,34 | 0,21 | 0,22 | 0,32 |
| 0,59 | 0,15 | 0,02 | 0,11 | 0,53 | 0,18 | 0,18 | 0,37 | 0,42 | 0,12 | 0,38 | 0,3  | 1    | 0,98 | 0,98 | 0,99 | 0,99 | 1    |
| 0,52 | 0,07 | 0    | 0,16 | 0,46 | 0,11 | 0,01 | 0,22 | 0,26 | 0    | 0,23 | 0,14 | 0,98 | 1    | 0,97 | 0,99 | 1    | 0,98 |
| 0,53 | 0,18 | 0,01 | 0,29 | 0,52 | 0,2  | 0,23 | 0,43 | 0,45 | 0,17 | 0,44 | 0,34 | 0,98 | 0,97 | 1    | 0,99 | 0,99 | 0,99 |
| 0,6  | 0,19 | 0,04 | 0,22 | 0,56 | 0,23 | 0,09 | 0,3  | 0,32 | 0,03 | 0,31 | 0,21 | 0,99 | 0,99 | 0,99 | 1    | 1    | 0,99 |
| 0,55 | 0,13 | 0    | 0,19 | 0,5  | 0,16 | 0,1  | 0,3  | 0,33 | 0,04 | 0,31 | 0,22 | 0,99 | 1    | 0,99 | 1    | 1    | 0,99 |
| 0,56 | 0,16 | 0,02 | 0,14 | 0,52 | 0,19 | 0,21 | 0,4  | 0,44 | 0,15 | 0,41 | 0,32 | 1    | 3,98 | 0,99 | 0,99 | 0,99 | 1    |

Рис. 7. Матрица корреляций  $P$ ,  $k = 4$ . Красным цветом показана слабая корреляция между четвёртым постом первой группы и остальными постами данной группы

увеличении числа  $k$  различия групп оказываются существеннее: при  $k = 3$  можно разделить посты на группы, предположив, что связь между постами существует при коэффициенте корреляции между соответствующими постами векторами  $> 0.7$ , и отсутствует при меньшем.

Однако при дальнейшем увеличении числа  $k$  алгоритм может не находить связи постов, что видно на рисунке 7. При ранге  $k = 5, 6$  подобных ошибок становится ещё больше. Было решено, что в случае трёх тематик для данных постов наилучшим рангом является  $k = 3$ .

Далее рассмотрены варианты улучшения существующего алгоритма разбиения постов по тематикам.

## 2.4. Методы улучшения существующего алгоритма

Методы улучшения алгоритма были подразделены на 3 группы.

- Методы уменьшения размерности матрицы  $A$ :
  - стемминг данных;
  - удаление шумовых слов.
- Изменение веса слова в матрице  $A$ :
  - подсчёт количества слов в посте;
  - TF-IDF.
- Изменение алгоритма расчёта матрицы корреляций  $P$ :
  - использование корреляций Спирмена вместо подсчёта косинуса угла между векторами.

## 2.5. Методы уменьшения размерности матрицы $A$

Перед более подробным рассмотрением каждого метода в отдельности стоит заметить, что данная группа методов весьма важна, так как позволяет получить не только более качественные результаты, но и существенно уменьшить время и объём памяти, необходимые для работы алгоритмов сингулярного разложения и построения матрицы корреляций.

## 2.6. Стемминг

Первая идея, которая возникает после осознания необходимости доработки алгоритма — стемминг строк, — нахождение основы для заданного исходного слова.

Существует немало алгоритмов стемминга:

- алгоритмы поиска — поиск флексивной формы слова в таблице поиска;
- алгоритмы усечения окончаний, представляющие собой последовательность шагов по усечению окончания слова;
- алгоритмы усечения окончаний с дополнительными критериями, например, с заменой окончания;
- аффикс-стеммеры, которые также работают не только с суффиксами, но и с префиксами;

- прочие, такие как алгоритмы сопоставления, стохастические алгоритмы, гибридные алгоритмы и т. д.

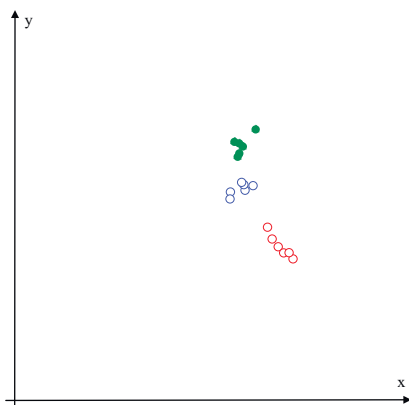


Рис. 8. Вектор-значения матрицы  $V^T$  при применении стемминга Портера

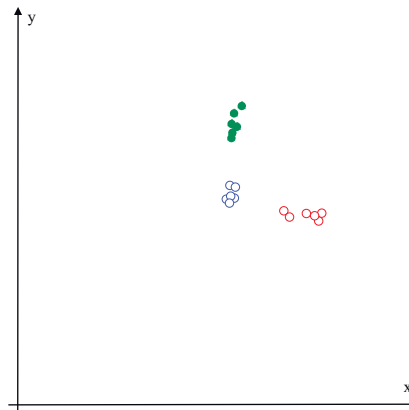


Рис. 9. Вектор-значения матрицы  $V^T$  при применении алгоритмов стемминга и удаления шумовых слов

Среди всего многообразия алгоритмов было решено остановиться на классическом Портеровском алгоритме усечения окончаний. Основная его идея заключается в том, что существует конечное количество словообразующих суффиксов. На каждом шаге алгоритма отсекается суффикс и оставшаяся часть проверяется на соответствие правилам языка. Если полученное слово удовлетворяет правилам — переход на следующий шаг, если нет — алгоритм выбирает другой суффикс для отсечения. Сначала отсекается максимальный формообразующий суффикс, затем — буква «и», потом словообразующие суффиксы, суффиксы превосходных форм, «ь» и одна из двух «н».

Среди плюсов алгоритма Портера — простота реализации, высокая скорость работы, отсутствие необходимости хранения каких-либо вспомогательных данных. К минусам же его работы можно отнести затруднённое получение правильной основы слова, отсутствие проверок на выпадающие и беглые гласные, отсутствие отсечений префиксов.

После применения алгоритма стемминга были получены следующие результаты, представленные на рисунке 8.

Из рисунка 8 видно, что посты формируют гораздо более чёткие группы, можно выделить не только группу постов про футбол, но и про кулинарию и органическую химию. Однако всё ещё может быть затруднительно чётко обозначить границы отдельных групп лишь по значениям матрицы  $V^T$ .

Стоит отметить, что при использовании стемминга Портера (рис. 10) группы постов хорошо различимы по матрице корреляций уже при ранге  $k = 2$ , однако, вновь наилучший результат достигается при использовании ранга  $k = 3$ .

## 2.7. Удаление шумовых слов

Следующая идея — удаление шумовых слов.

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,92 | 0,93 | 0,95 | 0,99 | 0,92 | 0,24 | 0,26 | 0,35 | 0,11 | 0,27 | 0,2  | 0,58 | 0,5  | 0,57 | 0,58 | 0,55 | 0,57 |
| 0,92 | 1    | 1    | 0,99 | 0,95 | 1    | 0,24 | 0,2  | 0,28 | 0,12 | 0,22 | 0,17 | 0,22 | 0,13 | 0,22 | 0,22 | 0,19 | 0,21 |
| 0,93 | 1    | 1    | 1    | 0,95 | 1    | 0,19 | 0,16 | 0,24 | 0,08 | 0,18 | 0,12 | 0,23 | 0,14 | 0,22 | 0,23 | 0,2  | 0,22 |
| 0,95 | 0,99 | 1    | 1    | 0,97 | 0,99 | 0,18 | 0,16 | 0,25 | 0,07 | 0,18 | 0,12 | 0,3  | 0,21 | 0,29 | 0,3  | 0,28 | 0,29 |
| 0,99 | 0,95 | 0,95 | 0,97 | 1    | 0,94 | 0,15 | 0,16 | 0,25 | 0,02 | 0,17 | 0,11 | 0,5  | 0,42 | 0,49 | 0,51 | 0,49 | 0,5  |
| 0,92 | 1    | 1    | 0,99 | 0,94 | 1    | 0,25 | 0,21 | 0,29 | 0,13 | 0,23 | 0,18 | 0,21 | 0,12 | 0,21 | 0,21 | 0,18 | 0,21 |
| 0,24 | 0,24 | 0,19 | 0,18 | 0,15 | 0,25 | 1    | 0,99 | 0,98 | 0,99 | 0,99 | 0,99 | 0,17 | 0,13 | 0,22 | 0,13 | 0,07 | 0,19 |
| 0,26 | 0,2  | 0,16 | 0,16 | 0,16 | 0,21 | 0,99 | 1    | 1    | 0,98 | 1    | 1    | 0,3  | 0,26 | 0,34 | 0,26 | 0,2  | 0,31 |
| 0,35 | 0,26 | 0,24 | 0,25 | 0,25 | 0,29 | 0,98 | 1    | 1    | 0,95 | 1    | 0,99 | 0,36 | 0,32 | 0,41 | 0,33 | 0,26 | 0,37 |
| 0,11 | 0,12 | 0,08 | 0,07 | 0,02 | 0,13 | 0,99 | 0,98 | 0,95 | 1    | 0,98 | 0,99 | 0,09 | 0,06 | 0,14 | 0,05 | 0    | 0,1  |
| 0,27 | 0,22 | 0,18 | 0,18 | 0,17 | 0,23 | 0,99 | 1    | 1    | 0,98 | 1    | 1    | 0,28 | 0,24 | 0,33 | 0,24 | 0,18 | 0,29 |
| 0,2  | 0,17 | 0,12 | 0,12 | 0,11 | 0,18 | 0,99 | 1    | 0,99 | 0,99 | 1    | 1    | 0,24 | 0,21 | 0,29 | 0,2  | 0,14 | 0,25 |
| 0,58 | 0,22 | 0,23 | 0,3  | 0,5  | 0,21 | 0,17 | 0,3  | 0,36 | 0,09 | 0,28 | 0,24 | 1    | 1    | 1    | 1    | 0,99 | 1    |
| 0,5  | 0,13 | 0,14 | 0,21 | 0,42 | 0,12 | 0,13 | 0,26 | 0,32 | 0,06 | 0,24 | 0,21 | 1    | 1    | 0,99 | 1    | 1    | 1    |
| 0,57 | 0,22 | 0,22 | 0,29 | 0,49 | 0,21 | 0,22 | 0,34 | 0,41 | 0,14 | 0,33 | 0,29 | 1    | 0,99 | 1    | 1    | 0,99 | 1    |
| 0,55 | 0,22 | 0,23 | 0,3  | 0,51 | 0,21 | 0,13 | 0,26 | 0,33 | 0,05 | 0,24 | 0,2  | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,55 | 0,19 | 0,2  | 0,28 | 0,49 | 0,18 | 0,07 | 0,2  | 0,26 | 0    | 0,18 | 0,14 | 0,99 | 1    | 0,99 | 1    | 1    | 0,99 |
| 0,57 | 0,21 | 0,22 | 0,29 | 0,5  | 0,21 | 0,19 | 0,31 | 0,37 | 0,1  | 0,29 | 0,25 | 1    | 1    | 1    | 1    | 0,99 | 1    |

Рис. 10. Матрица корреляции  $P$  при использовании стемминга Портера,  $k = 3$ 

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,9  | 0,91 | 0,92 | 0,98 | 0,87 | 0    | 0,05 | 0,1  | 0    | 0,03 | 0,03 | 0,5  | 0,38 | 0,44 | 0,45 | 0,5  | 0,46 |
| 0,9  | 1    | 1    | 1    | 0,97 | 1    | 0    | 0,02 | 0,09 | 0    | 0    | 0,02 | 0,09 | 0    | 0,02 | 0,02 | 0,08 | 0,03 |
| 0,91 | 1    | 1    | 1    | 0,97 | 0,99 | 0    | 0    | 0,07 | 0    | 1    | 0    | 0,11 | 0    | 0,04 | 0,05 | 0,1  | 0,06 |
| 0,92 | 1    | 1    | 1    | 0,97 | 0,99 | 0,02 | 0,04 | 0,1  | 0,01 | 0,01 | 0,04 | 0,13 | 0    | 0,06 | 0,06 | 0,12 | 0,07 |
| 0,98 | 0,97 | 0,97 | 0,97 | 1    | 0,94 | 0    | 0    | 0,05 | 0    | 0    | 0    | 0,33 | 0,2  | 0,27 | 0,26 | 0,33 | 0,29 |
| 0,87 | 1    | 0,99 | 0,99 | 0,94 | 1    | 0,04 | 0,04 | 0,11 | 0,03 | 0,01 | 0,05 | 0,01 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0,02 | 0    | 0,04 | 1    | 0,99 | 0,99 | 1    | 0,98 | 1    | 0,03 | 0,02 | 0,05 | 0    | 0    | 0    |
| 0,05 | 0,02 | 0    | 0,04 | 0    | 0,04 | 0,99 | 1    | 1    | 0,99 | 1    | 1    | 0,18 | 0,17 | 0,19 | 0,06 | 0,04 | 0,04 |
| 0,1  | 0,09 | 0,07 | 0,1  | 0,05 | 0,11 | 0,99 | 1    | 1    | 0,99 | 0,99 | 1    | 0,16 | 0,14 | 0,17 | 0,04 | 0,03 | 0,02 |
| 0    | 0    | 0    | 0,01 | 0    | 0,03 | 1    | 0,99 | 0,99 | 1    | 0,99 | 1    | 0,06 | 0,06 | 0,08 | 0    | 0    | 0    |
| 0,03 | 0    | 0    | 0,01 | 0    | 0,01 | 0,98 | 1    | 0,99 | 0,99 | 1    | 1    | 0,2  | 0,19 | 0,22 | 0,09 | 0,06 | 0,06 |
| 0,03 | 0,02 | 0    | 0,04 | 0    | 0,05 | 1    | 1    | 1    | 1    | 1    | 1    | 0,12 | 0,11 | 0,13 | 0    | 0    | 0    |
| 8,5  | 0,09 | 0,11 | 0,13 | 0,33 | 0,01 | 0,03 | 0,18 | 0,16 | 0,06 | 0,2  | 0,12 | 1    | 0,99 | 1    | 0,99 | 0,99 | 0,99 |
| 0,38 | 0    | 0    | 0    | 0,2  | 0    | 0,02 | 0,17 | 0,14 | 0,06 | 0,19 | 0,11 | 0,99 | 1    | 1    | 0,99 | 0,98 | 0,99 |
| 0,44 | 0,02 | 0,04 | 0,06 | 0,27 | 0    | 0,05 | 0,19 | 0,17 | 0,08 | 0,22 | 0,13 | 1    | 1    | 1    | 0,99 | 0,99 | 0,99 |
| 0,45 | 0,02 | 0,05 | 0,06 | 0,28 | 0    | 0    | 0,06 | 0,04 | 0    | 0,09 | 0    | 0,99 | 0,99 | 0,99 | 1    | 1    | 1    |
| 0,5  | 0,08 | 0,1  | 0,12 | 0,33 | 0    | 0    | 0,04 | 0,03 | 0    | 0,06 | 0    | 0,99 | 0,98 | 0,99 | 1    | 1    | 1    |
| 0,46 | 0,03 | 0,06 | 0,07 | 0,29 | 0    | 0    | 0,04 | 0,02 | 0    | 0,06 | 0    | 0,99 | 0,99 | 0,99 | 1    | 1    | 1    |

Рис. 11. Матрица корреляции  $P$  при использовании алгоритмов стемминга и удаления шумовых слов,  $k = 3$ 

Стоп-слова (шумовые слова) — слова, самостоятельно не несущие никакой смысловой нагрузки. Их можно разбить на две группы: общие и зависимые. К общим можно отнести предлоги, союзы, междометия, частицы и прочие служебные части речи. Зависимые стоп-слова зависят от контекста и достаточно трудны для учёта.

Зачастую шумовые слова можно выявить за счёт их небольшого размера: обычно их длина составляет не более 3-4 символов. Чтобы случайно не отсечь значимое слово, было решено ограничиться длиной слова в 2 символа после проведения стемминга. Однако есть и более длинные смысловые слова, например «почему», «который». Помимо проверки слова на длину было решено также фильтровать слова по словарю стоп-слов.

Видно, что на рисунке 9 посты формируют отчётливые группы, можно наблюдать чёткие границы между ними.

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0,03 | 1    | 0    | 0,01 | 0,03 | 0    | 0,01 | 0,01 | 0    | 0    |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,01 | 0,01 | 0,04 | 0    | 0,01 | 0,02 | 0,05 | 0,01 | 0,04 | 0,03 | 0,01 | 0,02 |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,01 | 0,03 | 0    | 0    | 0,01 | 0,05 | 0,01 | 0,04 | 0,03 | 0,01 | 0,02 |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0,03 | 0    | 0    | 0,01 | 0,12 | 0,08 | 0,1  | 0,1  | 0,07 | 0,08 |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0,02 | 0    | 1    | 0    | 0,09 | 0,05 | 0,08 | 0,07 | 0,05 | 0,06 |
| 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,02 | 0,02 | 11   | 0,02 | 0    | 0,03 | 0,05 | 0,01 | 0,03 | 0,03 | 0    | 0,01 |
| 0    | 0,01 | 1    | 0    | 0    | 0,02 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,03 | 0,06 | 0,06 | 0    | 0    | 0    |
| 0    | 0,01 | 0,01 | 0    | 0    | 0,02 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,07 | 0,1  | 0,1  | 0,04 | 0,02 | 0,02 |
| 0,03 | 0,04 | 0,03 | 0,03 | 0,02 | 0,05 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,05 | 0,08 | 0,08 | 0,02 | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0,02 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,04 | 0,06 | 0,07 | 0,01 | 0    | 0    |
| 0    | 0,01 | 0    | 0    | 0    | 0,02 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,06 | 0,09 | 0,09 | 0,03 | 0,01 | 0,01 |
| 0,01 | 0,02 | 0,01 | 0,01 | 0    | 0,03 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,06 | 0,08 | 0,09 | 0,03 | 0,01 | 0,01 |
| 0,03 | 0,05 | 0,05 | 0,12 | 0,09 | 0,05 | 0,03 | 0,07 | 0,05 | 0,04 | 0,06 | 0,06 | 0,06 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0    | 0,01 | 0,01 | 0,08 | 0,05 | 0,01 | 0,06 | 0,1  | 0,08 | 0,06 | 0,09 | 0,08 | 0,08 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,01 | 0,04 | 0,04 | 0,1  | 0,08 | 0,03 | 0,06 | 0,1  | 0,08 | 0,07 | 0,09 | 0,09 | 0,09 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,01 | 0,03 | 0,03 | 0,1  | 0,07 | 0,03 | 0    | 0,04 | 0,02 | 0,01 | 0,03 | 0,03 | 0,03 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0    | 0,01 | 0,01 | 0,07 | 0,05 | 0    | 0    | 0,02 | 0    | 0    | 0,01 | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0    | 0,02 | 0,02 | 0,08 | 0,06 | 0,01 | 0    | 0,02 | 0    | 0    | 0,01 | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    |

Рис. 13. Матрица корреляции  $P$  при подсчёте частоты слова в посте,  $k = 3$ 

При сравнении рисунков 10 и 11 можно заметить улучшение в матрице, вносимое данным методом.

## 2.8. Методы изменения веса слова в матрице $A$

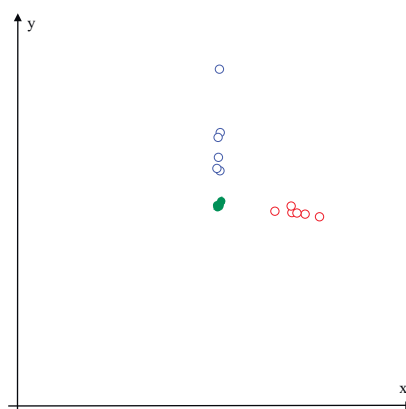
### 2.8.1. Подсчёт частоты слова в посте

При формировании матрицы  $A$ , отражающей присутствие слов в посте пользователя, может возникнуть вопрос, почему данная матрица состоит лишь из нулей и единиц. Возможно, качество алгоритма возрастёт, если использовать некую меру. Для начала было решено попробовать вместо единиц и нулей подсчитывать частоту слова в посте.

Видно (рисунки 12 и 13), что использование частоты слова в посте в качестве коэффициентов матрицы  $A$  приводит к существенному улучшению работы алгоритма — теперь корреляция для постов на одинаковую тему достигает единицы, для постов на разную — не превышает одной десятой.

### 2.8.2. TF-IDF

В качестве ещё одной меры было решено попробовать TF-IDF — статистическую меру, используемую для оценки важности слова в контексте документа, являющегося частью коллекции документов. Вес некоторого слова пропорционален количеству его употребления в документе, и обратно пропорционален

Рис. 12. Вектор-значения матрицы  $V^T$  при подсчёте частоты слова в посте, использовании алгоритмов стемминга и удаления шумовых слов

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,08 | 0,07 | 0    | 0,12 | 0,02 | 0,08 | 0,08 | 0,14 | 0,09 | 0    | 0    |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,07 | 0,06 | 0    | 0,11 | 0,02 | 0,1  | 0,09 | 0,15 | 0,11 | 0    | 0    |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,07 | 0,06 | 0    | 0,11 | 0,01 | 0,09 | 0,08 | 0,14 | 0,1  | 0    | 0    |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,08 | 0,07 | 0    | 0,12 | 0,03 | 0,15 | 0,14 | 0,2  | 0,16 | 0,06 | 0    |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,07 | 0,05 | 0    | 0,11 | 0,01 | 0,1  | 0,1  | 0,16 | 0,12 | 0,01 | 0,02 |
| 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,08 | 0,07 | 0    | 0,12 | 0,03 | 0,09 | 0,08 | 0,15 | 0,1  | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0,99 | 0,99 | 1    | 0,99 | 1    | 0,05 | 0,14 | 0,1  | 0,01 | 0    | 0    |
| 0,08 | 0,07 | 0,07 | 0,08 | 0,07 | 0,08 | 0,99 | 1    | 1    | 0,99 | 1    | 1    | 0,08 | 0,17 | 0,13 | 0,04 | 0    | 0,01 |
| 0,07 | 0,06 | 0,06 | 0,07 | 0,05 | 0,07 | 0,99 | 1    | 1    | 0,99 | 1    | 1    | 0,08 | 0,16 | 0,12 | 0,03 | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0,99 | 0,99 | 1    | 0,99 | 1    | 0,06 | 0,14 | 0,1  | 0,01 | 0    | 0    |
| 0,12 | 0,11 | 0,11 | 0,12 | 0,11 | 0,12 | 0,99 | 1    | 1    | 0,99 | 1    | 1    | 0,07 | 0,16 | 0,13 | 0,03 | 0    | 0,01 |
| 0,02 | 0,02 | 0,01 | 0,03 | 0,01 | 0,03 | 1    | 1    | 1    | 1    | 1    | 1    | 0,07 | 0,16 | 0,12 | 0,03 | 0    | 0    |
| 0,08 | 0,1  | 0,09 | 0,15 | 0,1  | 0,09 | 0,05 | 0,08 | 0,08 | 0,06 | 0,07 | 0,07 | 1    | 0,94 | 0,89 | 0,99 | 0,99 | 0,16 |
| 0,08 | 0,09 | 0,06 | 0,14 | 0,1  | 0,08 | 0,14 | 0,17 | 0,16 | 0,14 | 0,16 | 0,16 | 0,94 | 1    | 0,99 | 0,98 | 0,95 | 0,47 |
| 0,14 | 0,15 | 0,14 | 0,2  | 0,16 | 0,15 | 0,1  | 0,13 | 0,12 | 0,1  | 0,13 | 0,12 | 0,89 | 0,99 | 1    | 0,95 | 0,9  | 0,58 |
| 0,09 | 0,11 | 0,1  | 0,16 | 0,12 | 0,1  | 0,01 | 0,04 | 0,03 | 0,01 | 0,03 | 0,03 | 0,99 | 0,98 | 0,95 | 1    | 0,99 | 0,31 |
| 0    | 0    | 0    | 0,06 | 0,01 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,99 | 0,95 | 0,9  | 0,99 | 1    | 0,22 |
| 0    | 0    | 0    | 0    | 0,02 | 0    | 0    | 0,01 | 0    | 0    | 0,01 | 0    | 0,16 | 0,47 | 0,58 | 0,31 | 0,22 | 1    |

Рис. 14. Матрица корреляции  $P$  при использовании TF-IDF,  $k = 4$ 

частоте употребления слова в других документах коллекции. Большой вес в TF-IDF получают слова с высокой частотой в пределах конкретного документа и с низкой частотой использования в других.

Меру TF-IDF можно представить в виде произведения двух сомножителей:  $tf(t, d)$  и  $idf(t, d)$ , где

$$tf(t, d) = \frac{n_t}{\sum_k n_k},$$

$$idf(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|},$$

$t$  — слово,  $d$  — документ,  $D$  — совокупность документов. Во второй формуле не имеет значения выбор основания логарифма, так как его изменение приводит к изменению веса каждого слова на постоянный множитель.

При подсчёте матрицы корреляции выяснялось, что лучшие результаты при использовании меры TF-IDF получаются при  $k = 4$ .

Выяснилось, что получившиеся результаты всё же оказались хуже, чем при подсчёте частоты слова в посте, однако, существенно лучше, чем при простом учёте наличия слова в посте (рис. 14).

## 2.9. Использование корреляций Спирмена

Вместо подсчёта косинуса угла между двумя векторами для составления матрицы смежности было решено попробовать воспользоваться ранговыми корреляциями Спирмена.

Коэффициент корреляции Спирмена между двумя векторами вычисляется по формуле

$$\hat{\rho} = 1 - \frac{6 \sum (R_i - S_i)_t^2}{n(n^2 - 1)},$$

где  $R_i$  — ранг наблюдения  $x_i$  в векторе  $x$ ,  $S_i$  — ранг наблюдения  $y_i$  в векторе  $y$ .

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,58 | 0,66 | 0,6  | 0,97 | 0,56 | 0    | 0    | 0    | 0    | 0    | 0    | 0,02 | 0    | 0    | 0,09 | 0,23 | 0,15 |
| 0,58 | 1    | 0,95 | 0,99 | 0,67 | 0,99 | 0,17 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,66 | 0,95 | 1    | 0,94 | 0,75 | 0,92 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,6  | 0,99 | 0,94 | 1    | 0,68 | 0,98 | 0,16 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,97 | 0,67 | 0,75 | 0,68 | 1    | 0,65 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,14 | 0,06 |
| 0,56 | 0,99 | 0,92 | 0,98 | 0,65 | 1    | 0,21 | 0    | 0,01 | 0,02 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0,17 | 0    | 0,16 | 0    | 0,21 | 1    | 0,6  | 0,75 | 0,96 | 0,51 | 0,83 | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,6  | 1    | 0,79 | 0,74 | 0,96 | 0,87 | 0,1  | 0,21 | 0,17 | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0,01 | 0,75 | 0,79 | 1    | 0,79 | 0,66 | 0,9  | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0,02 | 0,96 | 0,74 | 0,79 | 1    | 0,65 | 0,93 | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,51 | 0,96 | 0,68 | 0,65 | 1    | 0,78 | 0,23 | 0,35 | 0,31 | 0,12 | 0    | 0,05 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,83 | 0,87 | 0,9  | 0,93 | 0,78 | 1    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,02 | 0    | 0    | 0    | 0    | 0    | 0    | 0,1  | 0    | 0    | 0,23 | 0    | 1    | 0,97 | 0,99 | 0,96 | 0,93 | 0,95 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,21 | 0    | 0    | 0,35 | 0    | 0,97 | 1    | 0,99 | 0,94 | 0,89 | 0,92 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,17 | 0    | 0    | 0,31 | 0    | 0,99 | 0,99 | 1    | 0,95 | 0,9  | 0,93 |
| 0,09 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,12 | 0    | 0,96 | 0,94 | 0,95 | 1    | 0,98 | 0,99 |
| 0,23 | 0    | 0    | 0    | 0,14 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,93 | 0,89 | 0,9  | 0,98 | 1    | 0,99 |
| 0,15 | 0    | 0    | 0    | 0,06 | 0    | 0    | 0    | 0    | 0    | 0,05 | 0    | 0,95 | 0,92 | 0,93 | 0,99 | 0,99 | 1    |

Рис. 15. Матрица корреляций  $P$ , составленная для матрицы наличия слова в посте при использовании корреляций Спирмена,  $k = 3$

Матрица корреляций с использованием корреляций Спирмена была составлена для трёх матриц: матрицы из 1 и 0, матрицы встречаемости слова в посте и матрицы из мер TF-IDF.

По получившимся матрицам корреляций нельзя однозначно сказать, являются ли корреляции Спирмена лучшим инструментом, чем подсчёт косинуса угла между векторам: они хорошо фильтруют «мусорные» связи, но также могут уменьшать и реальные связи (рисунки 15, 16 и 17).

## 2.10. Определение наличия у пользователя заданного предпочтения

Видно, что программа умеет разбивать предпочтения пользователей на группы, но как понять, что конкретная группа относится к данному предпочтению? Самый очевидный вариант — хранить некоторый объём текстовых данных, точно относящийся к этой теме.

Для проверки работоспособности предложенной теории было взято 5 текстов на футбольную тему, отличных от тех, что размещались в постах пользователя. Тексты были «подмешаны» в результат выполнения API-запроса `wall.get`.

Можно заметить (рисунки 19 и 18), что подмешанные тексты теперь составляют одну группу с постами пользователя на данную тему.

При написании функции проверки наличия у пользователя заданного предпочтения рассматривались только первые  $n$ -строк матрицы корреляции, где  $n$  — количество подмешанных текстов. Если более половины значений столбца, соответствующего посту  $T$ , были больше значения, говорящего о существовании связи между ним и проверочным словом (по умолчанию 0,7), то пост относился к данной тематике. Если доля постов пользователя/группы, относящихся к данной тематике, превышала заданное значение (по умолчанию 0,25), то принималось, что пользователь интересуется этой тематикой.

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,92 | 0,95 | 0,77 | 0,78 | 0,91 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,92 | 1    | 0,92 | 0,76 | 0,77 | 0,96 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,95 | 0,92 | 1    | 0,87 | 0,88 | 0,85 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0,77 | 0,76 | 0,87 | 1    | 0,99 | 0,69 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,03 |
| 0,78 | 0,77 | 0,66 | 0,99 | 1    | 0,69 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,01 |
| 0,91 | 0,96 | 0,85 | 0,69 | 0,69 | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0,95 | 0,87 | 0,99 | 0,96 | 0,94 | 0    | 0,2  | 0,15 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,95 | 1    | 0,86 | 0,96 | 0,99 | 0,97 | 0,13 | 0,38 | 0,34 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,87 | 0,86 | 1    | 0,85 | 0,84 | 0,93 | 0    | 0,04 | 0,06 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,99 | 0,96 | 0,85 | 1    | 0,96 | 0,95 | 0    | 0,26 | 0,2  | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,96 | 0,99 | 0,84 | 0,98 | 1    | 0,97 | 0,1  | 0,36 | 0,31 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,94 | 0,97 | 0,93 | 0,95 | 0,97 | 1    | 0,05 | 0,25 | 0,24 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,13 | 0    | 0    | 0,1  | 0,05 | 1    | 0,9  | 0,95 | 0,91 | 0,8  | 0,82 | 0,75 | 0,75 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,2  | 0,36 | 0,04 | 0,26 | 0,36 | 0,25 | 0,9  | 1    | 0,97 | 0,83 | 0,77 | 0,75 | 0,96 | 0,99 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,15 | 0,34 | 0,06 | 0,2  | 0,31 | 0,24 | 0,95 | 0,97 | 1    | 0,85 | 0,75 | 0,75 | 0,96 | 0,99 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,91 | 0,83 | 0,85 | 1    | 0,95 | 0,96 | 0,99 | 1    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,8  | 0,77 | 0,75 | 0,95 | 1    | 0,99 | 1    | 1    |
| 0    | 0    | 0    | 0,03 | 0,01 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,82 | 0,75 | 0,75 | 0,96 | 0,99 | 1    | 1    | 1    |

Рис. 16. Матрица корреляций  $P$ , составленная для матрицы встречаемостей слов в постах при использовании корреляций Спирмена,  $k = 3$

|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 0,99 | 1    | 0,97 | 0,98 | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,11 | 0    | 0    | 0    | 0    |
| 0,99 | 1    | 1    | 0,98 | 0,99 | 0,99 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,15 | 0    | 0    | 0    | 0    |
| 1    | 1    | 1    | 0,97 | 0,99 | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,15 | 0    | 0    | 0    | 0    |
| 0,97 | 0,98 | 0,97 | 1    | 0,97 | 0,97 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,22 | 0    | 0    | 0    | 0    |
| 0,98 | 0,99 | 0,99 | 0,97 | 1    | 0,98 | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,2  | 0    | 0    | 0    | 0    |
| 1    | 0,99 | 1    | 0,97 | 0,98 | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,11 | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0,58 | 0,62 | 1    | 0,52 | 0,79 | 0,03 | 0,28 | 0    | 0    | 0,16 | 0,09 | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,58 | 1    | 0,98 | 0,58 | 0,97 | 0,89 | 0,07 | 0,19 | 0,05 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,62 | 0,98 | 1    | 0,61 | 0,94 | 0,93 | 0,14 | 0,19 | 0,04 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 1    | 0,58 | 0,61 | 1    | 0,51 | 0,79 | 0,02 | 0,32 | 0,01 | 0    | 0,15 | 0,11 | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,52 | 0,97 | 0,94 | 0,51 | 1    | 0,81 | 0    | 0,04 | 0    | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,79 | 0,89 | 0,93 | 0,79 | 0,81 | 1    | 0,25 | 0,35 | 0,13 | 0    | 0    | 0    | 0    | 0    |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,03 | 0,07 | 0,14 | 0,02 | 0    | 0,25 | 1    | 0,72 | 0,8  | 0,68 | 0,42 | 0,05 | 0,25 | 0,25 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,28 | 0,19 | 0,19 | 0,32 | 0,04 | 0,35 | 0,72 | 1    | 0,93 | 0,55 | 0,3  | 0,25 | 0,28 | 0,28 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0,05 | 0,04 | 0,01 | 0    | 0,13 | 0,8  | 0,93 | 1    | 0,75 | 0,22 | 0,28 | 0,3  | 0,3  |
| 0,11 | 0,15 | 0,15 | 0,22 | 0,2  | 0,11 | 0    | 0    | 0    | 0    | 0    | 0    | 0,68 | 0,55 | 0,75 | 1    | 0,46 | 0,19 | 0,19 | 0,19 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,16 | 0    | 0    | 0,15 | 0    | 0    | 0,42 | 0,3  | 0,22 | 0,46 | 1    | 0,19 | 0,19 | 0,19 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,09 | 0    | 0    | 0,11 | 0    | 0    | 0,05 | 0,25 | 0,28 | 0,3  | 0,19 | 0,19 | 0,19 | 0,19 |

Рис. 17. Матрица корреляций  $P$  для матрицы из мер TF-IDF при использовании корреляций Спирмена,  $k = 4$



|      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,01 | 0,03 | 0    | 0    | 0,01 | 0,06 | 0,03 | 0,04 | 0,05 | 0,02 | 0,04 |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0    | 0,03 | 0    | 0    | 0,01 | 0,03 | 0,01 | 0,03 | 0,03 | 0    | 0,02 |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,01 | 0,04 | 0    | 0    | 0,02 | 0,05 | 0,02 | 0,03 | 0,03 | 0,01 | 0,02 |
| 0,99 | 0,99 | 0,99 | 1    | 0,98 | 0,99 | 0,99 | 0,99 | 1    | 0,99 | 0,99 | 0,12 | 0,14 | 0,16 | 0,12 | 0,12 | 0,14 | 0,11 | 0,08 | 0,09 | 0,09 | 0,06 | 0,08 |
| 1    | 1    | 1    | 0,98 | 1    | 1    | 1    | 1    | 0,99 | 1    | 1    | 0    | 0    | 0    | 0    | 0    | 0    | 0,02 | 0    | 0    | 0    | 0    | 0    |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 0,99 | 1    | 1    | 0,01 | 0,02 | 0,04 | 0    | 0    | 0,02 | 0,03 | 0    | 0,01 | 0,01 | 0    |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,02 | 0,04 | 0    | 0    | 0,02 | 0,05 | 0,02 | 0,03 | 0,03 | 0    | 0,02 |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0    | 0,01 | 0,03 | 0    | 0    | 0,01 | 0,05 | 0,02 | 0,03 | 0,03 | 0    | 0,02 |
| 1    | 1    | 1    | 1    | 0,99 | 0,99 | 1    | 1    | 1    | 1    | 1    | 0,03 | 0,04 | 0,07 | 0,02 | 0,03 | 0,05 | 0,13 | 0,1  | 0,11 | 0,11 | 0,08 | 0,1  |
| 1    | 1    | 1    | 0,99 | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 0,01 | 0,02 | 0,05 | 0,01 | 0,01 | 0,03 | 0,19 | 0,06 | 0,07 | 0,07 | 0,05 | 0,06 |
| 1    | 1    | 1    | 0,99 | 1    | 2    | 1    | 1    | 1    | 1    | 1    | 0,01 | 0,02 | 0,05 | 0,01 | 0,01 | 0,03 | 0,04 | 0,01 | 0,02 | 0,03 | 0    | 0,01 |
| 0    | 0    | 0    | 0    | 0    | 0    | 0,01 | 0    | 0    | 0,03 | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    | 0,03 | 0,05 | 0,06 | 0    | 0    |
| 0,01 | 0    | 0,01 | 0,14 | 0    | 0,02 | 0,02 | 0,01 | 0,04 | 0,02 | 0,02 | 1    | 1    | 1    | 1    | 1    | 1    | 0,07 | 0,09 | 0,1  | 0,04 | 0,01 | 0,02 |
| 0,03 | 0,03 | 0,04 | 0,16 | 0    | 0,04 | 0,04 | 0,03 | 0,07 | 0,05 | 0,05 | 1    | 1    | 1    | 1    | 1    | 1    | 0,05 | 0,07 | 0,08 | 0,02 | 0    | 0    |
| 0    | 0    | 0    | 0,12 | 0    | 0    | 0    | 0    | 0,02 | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    | 0,03 | 0,06 | 0,07 | 0,01 | 0    | 0    |
| 0    | 0    | 0    | 0,12 | 0    | 0    | 0    | 0    | 0,03 | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    | 0,06 | 0,05 | 0,09 | 0,03 | 0,01 | 0,01 |
| 0,01 | 0,01 | 0,02 | 0,14 | 0    | 0,02 | 0,02 | 0,01 | 0,05 | 0,03 | 0,03 | 1    | 1    | 1    | 1    | 1    | 1    | 0,06 | 0,08 | 0,09 | 0,03 | 0    | 0,01 |
| 0,06 | 0,05 | 0,05 | 0,11 | 0,02 | 0,03 | 0,05 | 0,05 | 0,13 | 0,09 | 0,04 | 0,03 | 0,01 | 0,05 | 0,03 | 0,06 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,03 | 0,01 | 0,02 | 0,08 | 0    | 0    | 0,02 | 0,02 | 0,1  | 0,06 | 0,01 | 0,05 | 0,09 | 0,07 | 0,06 | 0,08 | 0,08 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,04 | 0,01 | 0,03 | 0,09 | 0    | 0,01 | 0,03 | 0,03 | 0,11 | 0,07 | 0,02 | 0,06 | 0,1  | 0,08 | 0,07 | 0,09 | 0,09 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,05 | 0,03 | 0,03 | 0,09 | 0    | 0,01 | 0,03 | 0,03 | 0,11 | 0,07 | 0,03 | 0    | 0,04 | 0,02 | 0,01 | 0,03 | 0,03 | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,02 | 0    | 0,01 | 0,06 | 0    | 0    | 0    | 0    | 0,08 | 0,05 | 0    | 0    | 0,01 | 0    | 0    | 0,01 | 0    | 1    | 1    | 1    | 1    | 1    | 1    |
| 0,01 | 0,02 | 0    | 0,08 | 0    | 0    | 0,02 | 0,02 | 0,1  | 0,06 | 0,01 | 0    | 0,02 | 0    | 0    | 0,01 | 0,01 | 1    | 1    | 1    | 1    | 1    | 1    |

Рис. 18. Матрица корреляций при использовании матрицы частоты слова в посте. Первые пять столбцов и строк соответствуют подмешанным текстам

Из приведённого примера видно, что, храня некий набор данных, гарантированно относящийся к некоей тематике, можно проверять существование у пользователя постов на данную тематику и выявлять его заинтересованность ею.

### 2.11. Выбор ранга $k$ для групп с примерно равным количеством постов

Одним из недостатков метода латентно-семантического анализа является отсутствие четкого правила выбора ранга  $k$ . У различных авторов на этот счёт разное мнение: так, ранее упоминалось, что, например, в работе А. Д. Хомоненко и С.А. Краснова показывается, что для разделения двух групп наилучшим рангом является  $k = 2$  [4]. Было решено подобрать коэффициент эмпирически.

В ходе экспериментов обнаружено, что данный коэффициент зависит от количества различных тематик в постах пользователя/группы: так, для двух различных тематик наилучший  $k = 2$ , для трёх — 3, для четырёх — 4. Сложно сказать, можно ли экстраполировать полученные результаты на большее число тематик, но можно предположить, что при их малом количестве и при примерно одинаковом количестве постов в каждой группе, наилучший ранг  $k$  равен количеству различных тематик.

В случае же разного количества постов в различных тематиках было замечено, что тематики, для которых количество текстов невелико, оказывают значительно меньшее влияние на выбор  $k$ .

### 2.12. Проверка работы программы на реальных данных

Было решено посмотреть, как поведёт себя алгоритм поиска не на синтезированных данных, а на реальных. Для эксперимента выбрана группа, занимающаяся выкладыванием новостей на тему дорожно-транспортных происшествий,

откуда было взято 50 постов, и несколько прочих групп, откуда было взято ещё 50. Результаты запросов к VK API смешивались, и программе предстояло отличить посты, связанные с ДТП, от прочих.

Для выполнения этого задания в запрос было также «подмешано» 5 проверочных текстов на тему ДТП.

На данном этапе лучший результат показала матрица, состоящая из TF-IDF мер. Так, для вышеприведённого случая она дала только 2 ложноположительных результата и ни одного ложноотрицательного. Результаты работы программы можно ещё улучшить, увеличив количество «подмешанных» текстов. Однако на данном этапе вновь возникла проблема выбора ранга  $k$ . Был найден наилучший  $k$  для 15, 25, 35, 45, 55, 75 и 105 запросов (где в каждом случае 5 запросов являлись «примесью», а остальные — в равных частях посты на тему ДТП и др. посты) и составлен график

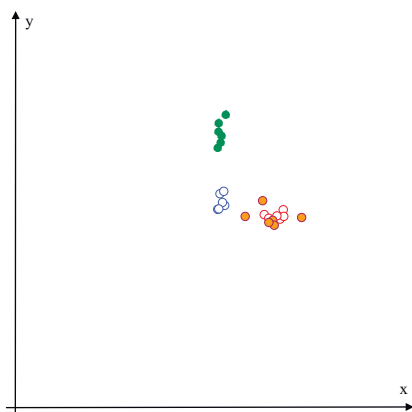


Рис. 19. Поиск постов с заданной тематикой с использованием матрицы из 0 и 1

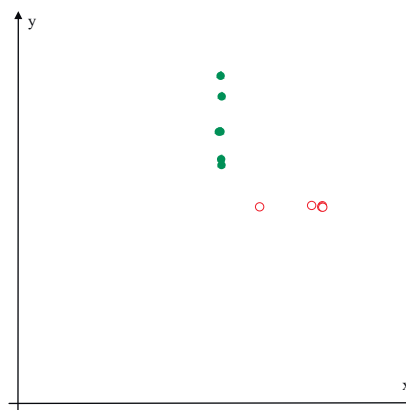


Рис. 20. Поиск по видеофайлам

По полученным результатам было замечено, что наименьшая ошибка аппроксимации возникает, если предположить, что зависимость наилучшего  $k$  от количества постов — логарифмическая (рисунок 21).

Данную зависимость можно выразить уравнением:  $y = 7.06535 \ln(0.10906x)$ , или приближённо:  $y = 7 \ln(0.1x)$ .

### 3. Разработка части приложения, определяющей предпочтения пользователей по размещённым аудио- и видеоматериалам

При разработке программы было выдвинуто предположение, что определять предпочтения пользователя можно не только по постам, но также по видео- и аудиоматериалам. В ходе разработки данной части программы использовались следующие API методы:

- `audio.get` — возвращает список аудиозаписей пользователя;
- `video.get` — возвращает информацию о видеозаписях.

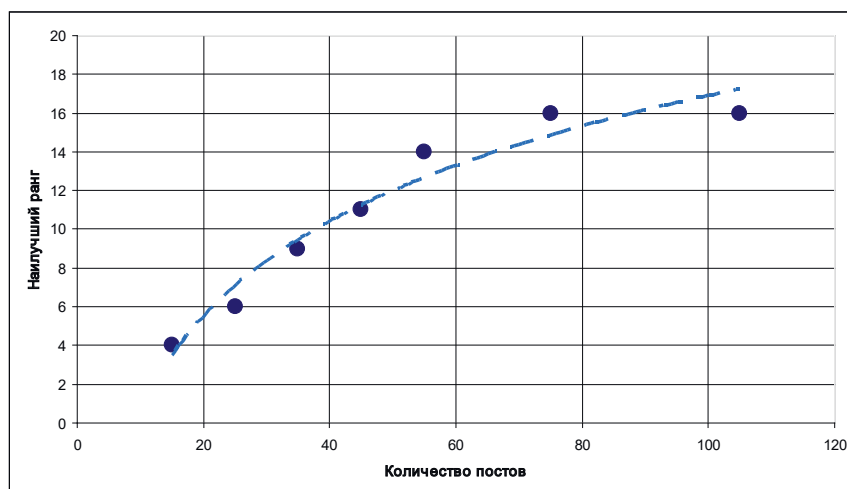


Рис. 21. Зависимость наилучшего ранга  $k$  от количества постов. Пунктирной линией показана предполагаемая логарифмическая зависимость

Полученные результаты приводились к соответствующим классам Video и Audio, откуда затем извлекались строки title — название и description — описание.

На основе вышеперечисленных API-методов и приведённой логики были созданы следующие функции:

- `Set<Video> getVideoById(Integer owner_id)` — выполняет поиск видеозаписей пользователя или группы;
- `Set<Audio> getAudioById(Integer owner_id)` — выполняет поиск аудиозаписей пользователя или группы.

Далее проводилось сингулярное разложение, стемминг и удаление шумовых слов по тем же алгоритмам, что и для записей на стене пользователя.

Работа функции была протестирована на следующем наборе данных:

- 5 видео на тему кулинарии;
- 5 видео на тему футбола.

Как видно из рисунка 20, результат выполнения функции чётко разбивается на 2 группы.

Стоит отметить, что данный метод поиска предпочтений пользователя не является эффективным из-за коротких названий видео- и аудиофайлов и очень редкого наличия описания к ним. Возможно, в данном случае намного эффективнее применять поиск по ключевым интересующим словам.

Также стоит обратить внимание на то, что с 16 декабря 2016 года руководство «ВКонтакте» отключило доступ к публичному музыкальному API сторонним приложениям, поэтому поиск аудиопредпочтений конкретного пользователя данным, равно как и другими методами, становится затруднительным, а то и вовсе невозможным.

## Заключение

В ходе данной работы были показаны этапы разработки приложения, осуществляющего поиск групп и пользователей в социальной сети «ВКонтакте», размещающих запрещённые материалы, а также проанализированы различные алгоритмы такого поиска, их эффективность. Наилучшим методом поиска был признан метод латентно-семантического анализа с использованием алгоритмов стемминга, фильтрации стоп-слов, TF-IDF мер и корреляций Спирмена, при выборе ранга матрицы  $\Sigma$  в сингулярном разложении  $k = 7 \ln(0.1x)$ , где  $x$  — количество постов.

Также хотелось бы отметить, что работа данной программы в социальной сети «ВКонтакте» часто бывает затруднена: нередко у пользователей отсутствуют посты или присутствуют в небольшом количестве, посты многих пользователей состоят из фотографий с краткой подписью или видеозаписей, из которых сложно извлечь сколь угодно значимую информацию, часто текст накладывают поверх картинки в графическом редакторе и т. п. Особый интерес для проведения поисков запрещённых материалов представляет социальная сеть «Твиттер», так как её особенности располагают к получению хороших результатов работы программы: твит ограничен 140 символами, что ограничивает количество слов, а, следовательно, и размер матрицы  $A$ ; большинство сообщений содержат текст, а не ссылки или изображения.

Настройка программы на поиск суицидальных групп «Синий кит» осуществляется добавлением в классификацию текстов заданий этих групп, находящихся в открытом доступе на новостных ресурсах<sup>1 2</sup>.

## Благодарности

Выражаем искреннюю признательность Надежде Фёдоровне Богаченко и Ольге Анатольевне Вишняковой за ценные советы, позволившие улучшить изложение результатов.

## ЛИТЕРАТУРА

1. Социально-сетевая жизнь [Электронный ресурс]. 2015. URL: [http://romir.ru/studies/670\\_1432155600/](http://romir.ru/studies/670_1432155600/) (дата обращения: 21.01.2017).
2. Антиэкстремизм в виртуальной России в 2014–2015 годы [Электронный ресурс]. 2016. URL: <http://www.sova-center.ru/racism-xenophobia/publications/2016/06/d34913/> (дата обращения: 02.02.2017).

---

<sup>1</sup>Игра синий кит все задания с 1 по 50 — полная информация про синего кита (подробный материал) [Электронный ресурс] // POLIKSAL. URL: <http://poliksal.ru/novosti-v-mire/180727-igra-siniy-kit-vse-zadaniya-s-1-po-50-polnaya-informaciya-pro-sinego-kita-podrobnyy-material.html> (дата обращения: 28.08.2017)

<sup>2</sup>Появился список заданий из суицидальных игр для подростков [Электронный ресурс] // СУПЕРОМСК. URL: [http://superomsk.ru/news/44483-poyavilsya\\_spisok\\_zadaniy\\_iz\\_suitsidalnx\\_igr\\_dlya/](http://superomsk.ru/news/44483-poyavilsya_spisok_zadaniy_iz_suitsidalnx_igr_dlya/) (дата обращения: 28.08.2017)

3. Отключение публичного API для аудио [Электронный ресурс]. 2017. URL: [https://vk.com/dev/audio\\_api/](https://vk.com/dev/audio_api/) (дата обращения: 22.01.2017).
4. Хомоненко А.Д., Краснов С.А. Применение метода латентно-семантического анализа для автоматической рубрикации документов // Известия Петербургского университета путей сообщения. 2012. № 2(31). С. 124–132.
5. Латентно-семантический анализ [Электронный ресурс]. 2010. URL: <https://habrahabr.ru/post/110078/> (дата обращения: 22.01.2017).
6. Алгоритм LSA для поиска похожих документов [Электронный ресурс]. 2014. URL: <http://netpeak.net/ru/blog/algorithm-lsa-dlya-poiska-pohozhih-dokumentov/> (дата обращения: 31.01.2017).
7. Осипова Ю.А., Лавров Д.Н. Применение кластерного анализа методом k-средних для классификации текстов научной направленности // Математические структуры и моделирование. № 3(43). 2017.

## IDENTIFICATION OF SUICIDAL GROUPS AND VIOLATORS OF COPYRIGHT IN SOCIAL NETWORKS

**A.V. Kostylev**

Student, e-mail: kostart\_94@mail.ru

**D.N. Lavrov**

Ph.D. (Eng.), Associate Professor, e-mail: dmitry.lavrov72@gmail.com

**A.K. Guts**

Dr.Sc. (Phys.-Math.), Professor, e-mail: aguts@mail.ru

Dostoevsky Omsk State University

**Abstract.** In this paper, the task is to develop software that allows you to automate the search for groups and users in social networks that host suicidal materials and materials that violate copyright in the network. As a social network for exploring the possibility of such a search, the social network “VKontakte” — the largest social network of the CIS was chosen. The latent semantic analysis algorithm was based on the developed software. The main variants of the algorithm and the most adequately working on real data version is selected.

**Keywords:** Classification of texts, latent-semantic analysis, copyright, offenses, social networks, suicidal groups.

*Дата поступления в редакцию: 29.06.2017*

## Авторам

### Предоставляемые данные и документы

Автор предоставляет в редакцию:

- рукопись статьи в формате  $\text{\LaTeX}$  (см. требования к оформлению);
- список из трёх экспертов по тематике статьи, давших согласие написать рецензию на представленную работу<sup>3</sup>;
- экспертное заключение о возможности открытого опубликования.

### Лицензирование

Согласно ГК РФ ст. 1286 лицензионный договор с автором для публикации в периодических изданиях может быть заключён в устной форме. Сам факт получения рукописи статьи редколлегией журнала «Математические структуры и моделирование» является акцептом (принятием) лицензионного договора.

Все статьи в журнале «Математические структуры и моделирование» публикуются под лицензией Creative Commons Attribution 4.0 International (CC-BY). Текст лицензии находится по адресу <https://creativecommons.org/licenses/by/4.0/legalcode>.

### Требования к оформлению рукописи

К публикации принимаются рукописи объёмом не более 16 страниц.

Авторам необходимо предоставить следующую информацию на русском и английском языках:

- название статьи;
- список авторов с указанием
  - фамилии, имени и отчества,
  - учёного звания,
  - учёной степени,
  - должности,
  - места работы или учёбы,
  - действующего адреса электронной почты;
- аннотация (абстракт) объёмом от 100 до 250 слов;
- список ключевых слов.

Автор также указывает УДК (универсальный десятичный код) статьи. Его можно подобрать по тематике статьи в справочнике <http://msm.univer.omsk.su/udc/>.

Библиографические ссылки оформляются согласно ГОСТ 7.0.5–2008.

Рукопись статьи представляется в редакцию по электронной почте в двух форматах pdf и tex. Статья должна быть набрана с использованием макропакета  $\text{\LaTeX}$ и стиля msmb.cls, предоставляемого редакцией <http://msm.univer.omsk.su/files/msmb.zip>. Рекомендуется установить компилятор  $\text{\LaTeX}$ , так как именно им пользуются в редакции.

Отклонения в оформлении рукописи от приведённых правил позволяют редколлегии принять решение о снятии статьи с публикации. Статья может быть отклонена по причинам несоответствия тематике журнала или в связи с низким уровнем качества научного исследования.

В статье запрещается переопределять стандартные команды и окружения.

Нумеруемые формулы необходимо выделять в отдельную строку.

Нумерация только арабскими цифрами в порядке возрастания с единицы. Нумеровать следует только те формулы, на которые в тексте имеются ссылки.

---

<sup>3</sup>Необходимы полные данные экспертов (место работы, учёная степень, должность), с указанием способа связи с ними (e-mail, телефон). Редколлегия может обратиться к одному из экспертов из предложенного списка с просьбой написать рецензию или может назначить рецензента из собственного списка.

Запрещается использовать в формулах буквы русского алфавита. Если без них никак не обойтись, то следует использовать команду `\mbox{...}`.

Все рисунки и таблицы должны иметь подпись, оформленную с помощью команды `\caption{...}`.

Файлы с рисунками необходимо представить в формате PDF или EPS (использовать редакторы векторной графики типа InkScape, Adobe Illustrator или Corel Draw).

Используйте стандартные команды переключения на готический, каллиграфический и ажурный шрифты: `\mathfrak`, `\mathcal` и `\mathbb`.

Не допускается заканчивать статью рисунком или таблицей.

В списке литературы обязательно указание следующих данных: для книг — фамилии и инициалы авторов, название книги, место издания, издательство, год издания, количество страниц; для статей — фамилии и инициалы авторов, название статьи, название журнала, год издания, том, номер (выпуск), страницы начала и конца статьи (для депонированных статей обязательно указать номер регистрации).

Кавычки в русском тексте («абвгд») должны быть угловыми, в английском — прямыми верхними кавычками ("abcdeг" или "abcdeг").

Обязательна расшифровка сокращений при первом вхождении термина. Например: ... искусственный интеллект (ИИ)...

## Порядок рецензирования

Первичная экспертиза проводится главным редактором (заместителем главного редактора). При первичной экспертизе оценивается соответствие статьи тематике журнала, правилам оформления и требованиям, установленным редакцией журнала к научным публикациям.

Все статьи, поступившие в редакцию научного журнала «Математические структуры и моделирование», проходят через институт рецензирования.

Рецензент выбирается главным редактором журнала из числа членов редколлегии или ведущих специалистов по профилю данной работы.

Рецензенты уведомляются о том, что присланные им рукописи являются частной собственностью авторов и относятся к сведениям, не подлежащим разглашению. Рецензентам не разрешается делать копии статей для своих нужд.

Срок для написания рецензии устанавливается по согласованию с рецензентом.

Рецензия должна раскрывать актуальность представленного материала, степень научной новизны исследования, определять соответствие предлагаемого к публикации текста общему профилю издания и стиль изложения.

Рецензент выносит заключение о возможности опубликования статьи: «рекомендуется», «рекомендуется с учётом исправления замечаний, отмеченных рецензентом» или «не рекомендуется». В случае отрицательной рецензии редакция направляет автору мотивированный отказ, заверенный главным редактором или его заместителем.

В случае несогласия с мнением рецензента автор статьи имеет право предоставить аргументированный ответ в редакцию журнала. Статья может быть направлена на повторное рецензирование, либо на согласование в редакционную коллегию.

При наличии в рецензии рекомендаций по исправлению и доработке статьи автору направляется текст рецензии с предложением учесть их при подготовке нового варианта статьи или аргументированно (частично или полностью) их опровергнуть. Доработанная (переработанная) автором статья повторно направляется на рецензирование и рассматривается в общем порядке. В этом случае датой поступления в редакцию считается дата возвращения доработанной статьи.

После принятия редколлегией решения о допуске статьи к публикации автор информируется об этом и указываются сроки публикации.

Оригиналы рецензий хранятся в редакции в течение пяти лет.

### **Авторская этика**

Авторы публикаций должны гарантировать, что в список авторов включены только лица, соответствующие критериям авторства (лица, внёсшие значительный вклад в работу), и что заслуживающие авторства исследователи не исключены из списка авторов.

Должны работать вместе с редакторами или издателями для скорейшего исправления своих работ в случае обнаружения в них ошибок или упущений после публикации.

Обязаны незамедлительно уведомлять редакцию в случае обнаружения ошибки в любой поданной ими на публикацию, принятой для публикации или уже опубликованной работе.

Не вправе копировать из других публикаций ссылки на работы, с которыми они сами не ознакомились; цитаты и ссылки на другие работы должны быть точными и оформленными в соответствии с предъявляемыми требованиями.

Должны ссылаться максимально правильно и точно на имеющие отношение к публикации предыдущие работы как других исследователей, так и самих авторов, обращаясь, прежде всего к первоисточнику; дословное воспроизведение собственных работ и их перефразирование неприемлемы, они могут быть использованы лишь в качестве основы для новых выводов.

Необходимо указывать авторство данных, текста, рисунков и идей, которые автор получил из других источников — они не должны представляться, как принадлежащие автору публикации; прямые цитаты из работ других исследователей должны выделяться кавычками и соответствующей ссылкой.

Должны соблюдать нормы законодательства о защите авторских прав; материалы, защищённые авторским правом (например, таблицы, цифры или крупные цитаты), могут воспроизводиться только с разрешения их владельцев.

### **Памятка для перевода должностей, учёных степеней и званий на английский язык**

Профессор = Professor

Доцент = Associate Professor

Старший преподаватель = Assistant Professor

Преподаватель = Instructor

Ассистент = Instructor

Аспирант = Postgraduate Student или Ph.D. Student

Соискатель = Ph.D. Doctoral Candidate

Магистрант = Master's Degree Student

Студент = Student

д.ф.-м.н. = Dr.Sc. (Phys.-Math.)

к.ф.-м.н. = Ph.D. (Phys.-Math.)

д.т.н. = Dr.Sc. (Eng.)

к.т.н. = Ph.D. (Eng.)

Инженер-программист = Software Engineer

Старший/младший научный сотрудник = Senior/Junior Scientist Researcher

### **Электронная почта для отправки статей**

lavrov@omsu.ru — зам. главного редактора (ответственный за выпуск) Д.Н. Лавров.



Научный журнал

# Математические структуры и моделирование

№3(43)

Главный редактор

**А.К. Гуц**

Зам. глав. ред., выпускающий редактор

**Д.Н. Лавров**

Зам. глав. ред., технический редактор

**Н.Ф. Богаченко**

Корректор:

**И.Н. Баловнева**

Проверка корректности перевода:

**Е.А. Илюшечкин**

**А.Н. Кабанов**

---

**Адрес научной редакции**

Россия, 644077, Омск, пр. Мира, 55А

Омский государственный университет

E-mail: [guts@omsu.ru](mailto:guts@omsu.ru), [lavrov@omsu.ru](mailto:lavrov@omsu.ru)

Электронная версия журнала:

<http://msm.univer.omsk.su>

<http://msm.omsu.ru>



---

Подписано в печать 18.09.2017. Формат 60 × 84 1/8.

Усл. печ. л. 20. Тираж 100 экз. Заказ № 120.

---

Отпечатано на полиграфической базе издательства ОмГУ им. Ф.М. Достоевского  
644077, г. Омск, пр. Мира, 55А

ISSN 2222-8772



9 772222 877005

17043 >

