

Министерство науки и высшего образования РФ

Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского

**МАТЕМАТИКА И ИТ — ВМЕСТЕ В ЦИФРОВОЕ БУДУЩЕЕ**

*Сборник трудов II Молодежной школы*

*Нижний Новгород, 24–28 апреля 2023 г.*

Нижний Новгород  
ННГУ им. Н.И. Лобачевского  
2023

УДК 004.942(06)+519.876.5(06)  
ББК 22.18я43  
М34

**М34 Математика и ИТ – вместе в цифровое будущее.** Сборник трудов II Молодежной школы (Н. Новгород, 24–28 апреля 2023 г.) / Под ред. К.А. Баркалова, М.А. Богомоловой. – Нижний Новгород: Нижегородский госуниверситет им. Н.И. Лобачевского, 2023. – 120 с.

*Отв. за выпуск Т.Г. Смирнова*

ISBN 978-5-91326-815-0

Сборник трудов участников конкурса научно-исследовательских проектов по математике и информационным технологиям II Молодежной школы «Математика и ИТ – вместе в цифровое будущее», состоявшейся 24–28 апреля 2023 г. на базе Нижегородского государственного университета им. Н.И. Лобачевского, содержит тезисы докладов и короткие статьи, тематика которых посвящена информационным и сквозным технологиям цифровой экономики, математическому моделированию сложных процессов и явлений, численным методам их исследования.

Молодежная школа является одним из мероприятий, проводимых в 2023 г. и приуроченных к празднованию 60-летия факультета вычислительной математики и кибернетики ННГУ им. Н.И. Лобачевского.

ISBN 978-5-91326-815-0

УДК 004.942(06)+519.876.5(06)  
ББК 22.18я43

© Коллектив авторов, 2023  
© ННГУ им. Н.И. Лобачевского, 2023

## Предисловие

24–28 апреля 2023 г. в Институте информационных технологий, математики и механики ННГУ им. Н. И. Лобачевского состоялась II Молодежная школа «Математика и ИТ — вместе в цифровое будущее». Традиционно школа прошла в online-формате. Участники школы встретились с учеными Университета Лобачевского и ведущими практиками из российских и международных IT-компаний: Сбер, YADRO, 1С, ООО «ПИРФ», ООО «Квантум ИНК», Центр искусственного интеллекта МТС и IT-академия Lad.

Молодежная школа является одним из мероприятий, проводимых в 2023 г. и приуроченных к празднованию 60-летия факультета вычислительной математики и кибернетики ННГУ им. Н.И. Лобачевского. Первый в стране факультет ВМК ННГУ был создан в 1963 г. по инициативе профессора Юрия Исааковича Неймарка. В 2015 г. факультет ВМК и механико-математический факультет были объединены в институт информационных технологий, математики и механики – признанный в стране и за рубежом научно-образовательным центром подготовки математиков и программистов высочайшей квалификации.

Программа молодежной школы включала следующие лекции:

- Надежда Дюкина (Университет Лобачевского, доцент института ИТММ): «Решение задач прочности и пластичности в пакете инженерного моделирования Логос»;
- Алексей Усков (Академия Lad, руководитель): «Какие технологии нужно знать разработчику, QA-инженеру, бизнес и системному аналитику, проджектору, чтобы попасть в IT-компанию»;
- Алексей Малафеев (Центр искусственного интеллекта МТС, руководитель группы NLP Platform): «ChatGPT. Что под капотом?»;
- Михаил Иванченко (Университет Лобачевского, зав. каф. прикладной математики института ИТММ) «Объясняемый искусственный интеллект»;
- Басель Дарвиш (1С, руководитель направления облачных технологий и сервисов): «Облачные технологии»;
- Алексей Стёпин (DreamPics, Lead software engineer): «Digital art и новые горизонты генерации изображений. Заглядываем под капот к MidJourney и Stable Diffusion»;
- Анна Хакимова (YADRO, инженер разработки математических алгоритмов): «Линейная алгебра: задачи, которые решают математики в инженерных компаниях»;
- Николай Ильин, Владимир Башев (ООО «ПИРФ») «Инновации в области развития операционных систем»;
- Юрий Баташев (ООО «Квантум ИНК», генеральный директор) «Кибериммунный подход к проектированию архитектуры систем»;
- Иосиф Мееров (Университет Лобачевского, доцент института ИТММ) «Высокопроизводительные вычисления в задачах финансовой математики»;
- Константин Клименко (Сбер, директор по продуктам лаборатории блокчейн): «Блокчейн и будущее финансов».

В рамках работы Молодежной школы дополнительно проводился конкурс научно-исследовательских проектов по математике и информационным технологиям, тематика которых охватывала следующие направления:

- суперкомпьютерные технологии;
- искусственный интеллект;
- машинное обучение и анализ данных;
- компьютерное зрение;
- робототехника;

- обработка естественных языков;
- виртуальная реальность;
- дополненная реальность;
- интернет вещей;
- большие данные;
- математическое моделирование;
- экспериментальная механика;
- компьютерная механика;
- численные методы;
- алгебра, геометрия, теория чисел;
- теория вероятностей и приложения;
- дискретная математика;
- дифференциальные уравнения.

Всего на молодежную школу поступило более 180 заявок от студентов из 20 регионов нашей страны. В рамках работы Молодежной школы проводился конкурс научно-исследовательских проектов. На конкурс поступило более 40 научно-исследовательских работ.

Из них победителями стали: Роман Майтак и Пётр Пылов (Кузбасский государственный технический университет им. Т.Ф. Горбачева) с проектом «Применение имитационной модели искусственного интеллекта для автоматизированной оптимизации процесса разработки угольных месторождений»; Рамиль Вольников (Нижегородский государственный технический университет им. Р.Е. Алексеева) с проектом «Оптимизация аэродинамического профиля. Крыло с изменяемым профилем».

Второе место заняла Алла Боброва (Институт информационных технологий, математики и механики Нижегородского государственного университета им. Н.И. Лобачевского) с проектом «Синхронизация волновых процессов в нейросетевых системах: возможности квантовых алгоритмов оценки фазы».

Третье место заняли Зульфия Мухаметрахимова (Башкирский государственный педагогический университет им. М. Акмуллы) с проектом «Асимптотика собственного значения оператора Шредингера с растущим потенциалом» и Владимир Привезенов (Нижегородский государственный технический университет им. Р.Е. Алексеева) с проектом «Техника обучения нейронной сети по распознаванию объекта на фото с использованием 3D-модели».

# Свойства классов $A_\alpha^\infty$ и $X_\alpha^\infty$ , аналитических в полуплоскости функций

В.А Беднаж, Д.С. Ермакова

Брянский государственный университет им. И.Г. Петровского

Первая глава посвящена описанию корневых множеств классов  $A_\alpha^\infty$  аналитических в круге функций, а именно, рассмотрены свойства нулей функций из классов Бергмана. Во второй главе исследуются свойства корневых множеств  $X_\alpha^\infty(G)$  классов аналитических в односвязных областях функций. Все научные результаты, полученные в работе, являются новыми и могут использоваться в теории функций комплексного переменного, в теории аппроксимации, в теории операторов и других разделах комплексного и функционального анализа.

*Ключевые слова:* корневые множества, единичный круг, нули функции, классы Бергмана.

## 1. Введение

Теория аналитических функций, являясь основным разделом комплексного анализа, имеет широкое приложение в теории аппроксимации, теории операторов, вещественном анализе, других отраслях математики и физики.

Одним из важнейших вопросов в изучении аналитических функций является вопрос описания корневых множеств.

Пусть  $D = \{z: |z| < 1\}$  – единичный круг на комплексной плоскости,  $H(D)$  – множество всех аналитических в  $D$  функций. В 1914 году Харди рассмотрел класс  $H^p$ ,  $p > 0$ , аналитических функций, удовлетворяющих условию

$$\sup_{0 < r < 1} \int_{-\pi}^{\pi} |f(re^{i\theta})|^p d\theta < +\infty.$$

На сегодняшний день хорошо изучены корневые множества функций класса  $H^p$ . Известно, что если функция  $f \in H^p(D)$ ,  $0 < p < +\infty$ , то нули  $\{z_k\}_{k=1}^{+\infty}$  функции подчиняются известному условию Бляшке

$$\sum_{k=1}^{+\infty} (1 - |z_k|) < +\infty.$$

Обозначим через

$$A_\alpha^p(D) = \{f \in H(D): \int_D |f(z)|^p (1 - |z|)^\alpha dm_2(z) < +\infty\},$$

где  $m_2$  – плоская мера Лебега,  $-1 < \alpha < +\infty$ .

В литературе классы  $A_\alpha^p$  получили название классов Бергмана, хотя впервые их рассмотрел М.М. Джрбашян в своих исследованиях 40-х годов XX века [6]. Значительно позже свойствами функций классов  $A_\alpha^p$  занимались Г. Шапиро, А. Шилдс, П. Дьюрен и многие другие. Интересный материал о нулях функций классов  $A_\alpha^p$  содержится в работе израильского математика Ч. Горовица [3], доказавшего, что если  $0 < p < +\infty$  и  $\alpha > \beta > -1$ , то корневые множества пространств  $A_\alpha^p(D)$  и  $A_\beta^p(D)$  различны.

В последние годы интенсивно продолжается изучение классов  $A_\alpha^p$  аналитических функций в единичном круге  $D$  [2].

Естественно, возникает вопрос о распространении этих результатов на произвольную односвязную область комплексной плоскости.

Классы аналитических функций в односвязной области  $G$  впервые были рассмотрены в 1928 году В.И. Смирновым.

Пусть функция  $w = \phi(z)$  осуществляет конформное отображение единичного круга  $D$  на односвязную область  $G$ , граница которой содержит более одной точки.

Рассмотрим пространство Бергмана в области  $G$ :

$$A_\alpha^p(G) = \{F \in H(G): \int_G |F(w)|^p (\text{dist}(w, \partial G))^\alpha dm_2(w) < +\infty\},$$

где  $\text{dist}(w, \partial G)$  – расстояние от точки  $w$  до границы области  $\partial G$ .

Однако до сих пор не получено полное описание корней функций из классов  $A_\alpha^p$ ,  $0 < p < +\infty$ , в случае произвольной односвязной области комплексной плоскости.

Исходя из вышесказанного, следует, что данное исследование посвящено актуальному направлению комплексного анализа.

Объект исследования. Объектом исследования данной работы являются классы аналитических в односвязных областях функций.

Цель и задачи исследования. Целью данной работы является исследование корневых множеств некоторых классов аналитических в односвязных областях функций.

Методы исследования. В работе используются методы комплексного и функционального анализа, а также более специальные методы, основанные на геометрической теории функций комплексного переменного.

## 2. Свойство корневых множеств функций классов $A_\alpha^\infty$ , где $\alpha > -1$ , на односвязной области комплексной плоскости

Пусть  $D = \{z: |z| < 1\}$  – единичный круг на комплексной плоскости,  $H(D)$  – множество всех аналитических в  $D$  функций.

Введем в рассмотрение класс функций

$$A_\alpha^\infty(D) = \left\{f \in H(D): |f(z)| \leq \frac{c_f}{(1-|z|)^\alpha}\right\},$$

где  $c_f$  – положительная постоянная, зависящая только от  $f$ ,  $\alpha > -1$ .

В работе Ф. А. Шамомяна [1] установлено, что нули аналитической функции можно выделить, не выходя из пространства  $A_\alpha^\infty(D)$ .

Обобщим этот результат на случай произвольной односвязной области комплексной плоскости.

Обозначим

$$A_\alpha^\infty(G) = \left\{f \in H(G): |F(w)| \leq \frac{c_F}{(\text{dist}(w, \partial G))^\alpha}\right\},$$

где  $\text{dist}(w, \partial G)$  – расстояние от точки  $w$  до границы области  $\partial G$ ,  $c_F$  – положительная постоянная, зависящая только от  $F$ ,  $\alpha > -1$ .

**Теорема 1.** Если функция  $f \in A_\alpha^\infty(D)$ ,  $\alpha > -1$ , и  $f(z_k) = 0$ ,  $k = 1, 2, \dots$ ,  $z_k \in D$ , тогда функция  $\frac{f}{\tilde{B}} \in A_\alpha^\infty(D)$ , где  $\tilde{B}(z, z_k)$  – бесконечное произведение вида

$$\tilde{B}(z, z_k) = \prod_{k=1}^{+\infty} (2 - b_k(z, z_k)) \cdot b_k(z, z_k), \quad z, z_k \in D.$$

**Теорема 2.** Пусть  $G$  – односвязная область на комплексной плоскости, граница которой содержит более одной точки. Если функция  $F \in A_\alpha^\infty(G)$ ,  $\alpha > -1$ , и  $F(w_k) = 0$ ,  $k = 1, 2, \dots$ ,  $w_k \in G$ , тогда функция  $\frac{F}{\tilde{B}} \in A_\alpha^\infty(G)$ , где  $\tilde{B}(w, w_k)$  – бесконечное произведение вида

$$\tilde{B}(w, w_k) = \prod_{k=1}^{+\infty} (2 - b_k(\psi(w), \psi(w_k))) \cdot b_k(\psi(w), \psi(w_k)), \quad w, w_k \in G,$$

$\psi(w)$  – функция, конформно отображающая область  $G$  на единичный круг  $D$ .

*Доказательство.* Так как  $F \in A_\alpha^\infty(G)$ , то  $|F(w)| \leq \frac{c_F}{(\text{dist}(w, \partial G))^\alpha}$ .

Конформно отображим единичный круг  $D$  на область с помощью функции  $w = \phi(z)$  [7, с. 113]:

$$|F(\phi(z))| \leq \frac{c_F}{(\text{dist}(\phi(z), \partial G))^\alpha}.$$

Функция  $F(\phi(z)) \in H(D)$ . Обозначим  $h(z) = F(\phi(z))$ ,  $z \in D$ , тогда

$$|h(z)| \leq \frac{c_F}{(\text{dist}(\phi(z), \partial G))^\alpha}.$$

Используя двойное неравенство, которое выводится из теоремы Кебе [5, с. 51]:

$$\frac{1}{4} \frac{\text{dist}(\phi(z), \partial G)}{1-|z|} \leq |\phi'(z)| \leq 4 \frac{\text{dist}(\phi(z), \partial G)}{1-|z|}, \quad (1.1)$$

получим:

$$|h(z)| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha \cdot |\phi'(z)|^\alpha}.$$

Функция  $\phi'(z)$  не обращается в нуль внутри единичного круга [7]. Рассмотрим функцию

$$f(z) = h(z) \cdot (\phi'(z))^\alpha, \quad (1.2)$$

аналитическую в единичном круге и отличную от тождественного нуля. Тогда последнее неравенство будет эквивалентно следующему:

$$\left| \frac{f(z)}{(\phi'(z))^\alpha} \right| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha \cdot |\phi'(z)|^\alpha}$$

или

$$|f(z)| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha},$$

то есть  $f \in A_\alpha^\infty(D)$ .

Имеем,  $F(w_k) = 0$  и  $w_k = \phi(z_k)$ ,  $k = 1, 2, \dots$ , то  $F(\phi(z_k)) = 0$ , то есть  $h(z_k) = 0$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ . Так как нули функции  $h(z)$  совпадают с нулями функции  $f(z) = h(z) \cdot (\phi'(z))^\alpha$ , то  $f(z_k) = 0$ ,  $k = 1, 2, \dots$

Итак, нами построена функция  $\frac{f}{B} \in A_\alpha^\infty(D)$ , не имеющая нулей в  $A_\alpha^\infty(D)$ .

Отобразим конформно область  $G$  на единичный круг с помощью функции  $z = \psi(w)$ , получим:

$$\left| \frac{f(\psi(w))}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq \frac{4^\alpha \cdot c_F}{(1-|\psi(w)|)^\alpha}.$$

Используя оценки (1.1), замечаем, что указанное неравенство эквивалентно следующему:

$$\left| \frac{f(\psi(w))}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq \frac{4^\alpha \cdot c_F \cdot |\phi'(\psi(w))|^\alpha \cdot 4^\alpha}{(\text{dist}(w, \partial G))^\alpha}$$

или

$$\left| \frac{f(\psi(w))}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq \frac{c_{F,\alpha} \cdot |\phi'(\psi(w))|^\alpha}{(\text{dist}(w, \partial G))^\alpha}.$$

Функция  $f(\psi(w)) \in H(G)$ ,  $w \in G$ . Учитывая (1.2), получим

$$\left| \frac{F(w)}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq \frac{c_{F,\alpha}}{(\text{dist}(w, \partial G))^\alpha},$$

то есть функция  $\frac{F}{\tilde{B}} \in A_\alpha^\infty(G)$ , где

$$\tilde{B}(w, w_k) = \prod_{k=1}^{+\infty} (2 - b_k(\psi(w), \psi(w_k))) \cdot b_k(\psi(w), \psi(w_k)), \quad w, w_k \in G.$$

Далее, так как  $\tilde{B}(z, z_k) \neq 0$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ , то есть  $\prod_{k=1}^{+\infty} (2 - b_k(z, z_k)) \cdot b_k(z, z_k) \neq 0$ . А это справедливо тогда и только тогда, когда  $b_k(z, z_k) \neq 0$ , то есть  $\frac{z_k}{|z_k|} \cdot \frac{z_k - z}{1 - \bar{z}_k \cdot z_k} \neq 0$ ,  $k = 1, 2, \dots$ . Следовательно,  $z \neq z_k$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ . Отсюда заключаем, что  $\tilde{B}(w, w_k) \neq 0$ .

Таким образом, мы выделили нули аналитической функции, не выходя из пространства  $A_\alpha^\infty(G)$ .

Теорема 2 доказана.

### 3. Характеристика корневых множеств класса $X_\alpha^\infty$ аналитических на произвольной односвязной области функций

В работе используя некоторые ограничения на функцию, конформно отображающую единичный круг на односвязную область комплексной плоскости, получена оценка нулей функции из класса  $X_\alpha^\infty(G)$ .

Пусть  $D = \{z: |z| < 1\}$  – единичный круг на комплексной плоскости,  $H(D)$  – множество всех аналитических в  $D$  функций.

Рассмотрим класс функций

$$X_\alpha^\infty(D) = \left\{ f \in H(D): \ln|f(z)| \leq \frac{c_f}{(1-|z|)^\alpha} \right\},$$

где  $c_f$  – положительная постоянная, зависящая только от  $f$ .

В работе Ф.А. Шамояна [1] установлена следующая теорема.

**Теорема 1.** Если функция  $f \in X_\alpha^\infty(D)$ ,  $0 < \alpha < 3$ , и  $f(z_k) = 0$ ,  $k = 1, 2, \dots$ ,  $z_k \in D$ , тогда

$$\left| \frac{f(z)}{\tilde{B}(z, z_k)} \right| \leq c_1 \exp\left(\frac{c_2}{(1-|z|)^\alpha}\right),$$

где  $\tilde{B}(z, z_k)$  – бесконечное произведение вида

$$\tilde{B}(z, z_k) = \prod_{k=1}^{+\infty} (2 - b_k(z, z_k)) \cdot b_k(z, z_k), \quad z, z_k \in D,$$

$c_1, c_2$  – некоторые положительные постоянные.

Мы обобщаем этот результат на случай произвольной односвязной области комплексной плоскости, используя методы работ [1] и [4].

Обозначим

$$X_\alpha^\infty(G) = \left\{ f \in H(G): \ln|F(w)| \leq \frac{c_F}{(\text{dist}(w, \partial G))^\alpha} \right\},$$

где  $\text{dist}(w, \partial G)$  – расстояние от точки  $w$  до границы области  $\partial G$ ,  $c_F$  – положительная постоянная, зависящая только от  $F$ .

Итак, справедлива

**Теорема 2.** Пусть  $G$  – односвязная область на комплексной плоскости, граница которой содержит более одной точки,  $\phi(z)$  – функция, конформно отображающая единичный круг  $D$  на область  $G$ ,  $\psi$  – обратная функция для  $\phi$ . Если функция  $F \in X_\alpha^\infty(G)$ ,  $0 < \alpha < 3$ ,  $F(w_k) = 0$ ,  $k = 1, 2, \dots$ ,  $w_k \in G$  и справедлива оценка

$$c_1 \leq |\phi'(z)| \leq c_2, \quad (2.1)$$

тогда функция  $\frac{F}{\tilde{B}} \in X_\alpha^\infty(G)$ , где  $\tilde{B}(w, w_k)$  – бесконечное произведение вида

$$\tilde{B}(w, w_k) = \prod_{k=1}^{+\infty} (2 - b_k(\psi(w), \psi(w_k))) \cdot b_k(\psi(w), \psi(w_k)), \quad w, w_k \in G,$$

$c_1, c_2$  – некоторые положительные постоянные.

*Доказательство.* Так как  $F \in X_\alpha^\infty(G)$ , то  $\ln|F(w)| \leq \frac{c_F}{(\text{dist}(w, \partial G))^\alpha}$ .

Используя теорему Б. Римана [2, с. 113], конформно отображим единичный круг  $D$  на область  $G$  с помощью функции  $w = \phi(z)$ :

$$\ln|F(\phi(z))| \leq \frac{c_F}{(\text{dist}(\phi(z), \partial G))^\alpha}.$$

Используя оценки [4], получим:

$$\ln|F(\phi(z))| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha \cdot |\phi'(z)|^\alpha}.$$

Функция  $F(\phi(z)) \in H(D)$ . Обозначим  $f(z) = F(\phi(z))$ ,  $z \in D$ , тогда

$$\ln|f(z)| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha \cdot |\phi'(z)|^\alpha}.$$

Учитывая левую оценку (2.1), приходим к неравенству

$$\ln|f(z)| \leq \frac{4^\alpha \cdot c_F}{(1-|z|)^\alpha \cdot c_1^\alpha} = \frac{c_f}{(1-|z|)^\alpha},$$

то есть  $f \in X_\alpha^\infty(D)$ .

Имеем,  $F(w_k) = 0$  и  $w_k = \phi(z_k)$ ,  $k = 1, 2, \dots$ , то  $F(\phi(z_k)) = 0$ , то есть  $f(z_k) = 0$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ .

Согласно Теореме 1 справедлива оценка

$$\left| \frac{f(z)}{\tilde{B}(z, z_k)} \right| \leq c_1 \exp\left(\frac{c_2}{(1-|z|)^\alpha}\right),$$

где  $\tilde{B}(z, z_k) = \prod_{k=1}^{+\infty} (2 - b_k(z, z_k)) \cdot b_k(z, z_k)$ ,  $z, z_k \in D$ .

Итак, нами построена функция  $\frac{f}{\tilde{B}} \in X_\alpha^\infty(D)$ , не имеющая нулей в  $X_\alpha^\infty(D)$ .

Отобразим конформно область  $G$  на единичный круг с помощью функции  $z = \psi(w)$ , получим:

$$\left| \frac{f(\psi(w))}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq c_1 \exp\left(\frac{4^\alpha \cdot c_f}{(1-|\psi(w)|)^\alpha \cdot c_1^\alpha}\right).$$

Функция  $f(\psi(w)) \in H(G)$ . Обозначим  $F(w) = f(\psi(w))$ ,  $w \in G$ , тогда

$$\left| \frac{F(w)}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq c_1 \exp\left(\frac{4^\alpha \cdot c_F}{(1-|\psi(w)|)^\alpha \cdot c_1^\alpha}\right).$$

Используя оценки [4], замечаем, что указанное неравенство эквивалентно следующему:

$$\left| \frac{F(w)}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq c_1 \exp\left(\frac{4^\alpha \cdot c_F \cdot |\phi'(\psi(w))|^\alpha}{c_1^\alpha \cdot (\text{dist}(w, \partial G))^\alpha}\right).$$

Учитывая правую оценку (2.1), приходим к неравенству

$$\left| \frac{F(w)}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq c_1 \exp\left(\frac{4^\alpha \cdot c_F \cdot c_2^\alpha}{c_1^\alpha \cdot (\text{dist}(w, \partial G))^\alpha}\right) = c_1 \exp\left(\frac{c_{\alpha, F}}{(\text{dist}(w, \partial G))^\alpha}\right)$$

или

$$\ln \left| \frac{F(w)}{\tilde{B}(\psi(w), \psi(w_k))} \right| \leq \frac{c_F}{(\text{dist}(w, \partial G))^\alpha},$$

то есть функция  $\frac{F}{\tilde{B}} \in X_\alpha^\infty(G)$ , где

$$\tilde{B}(w, w_k) = \prod_{k=1}^{+\infty} (2 - b_k(\psi(w), \psi(w_k))) \cdot b_k(\psi(w), \psi(w_k)), \quad w, w_k \in G.$$

Так как  $\tilde{B}(z, z_k) \neq 0$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ , то есть  $\prod_{k=1}^{+\infty} (2 - b_k(z, z_k)) \cdot b_k(z, z_k) \neq 0$ .

А это справедливо тогда и только тогда, когда  $b_k(z, z_k) \neq 0$ , то есть  $\frac{\bar{z}_k}{|z_k|} \cdot \frac{z_k - z}{1 - \bar{z}_k \cdot z_k} \neq 0$ ,  $k = 1, 2, \dots$

Следовательно,  $z \neq z_k$ ,  $z_k = \psi(w_k)$ ,  $k = 1, 2, \dots$ . Отсюда заключаем, что  $\tilde{B}(w, w_k) \neq 0$ .

Таким образом, мы выделили нули аналитической функции, не выходя из пространства  $X_\alpha^\infty(G)$ .

Теорема 2 доказана.

## Литература

1. Djrbashian A.E. and Shamoyan F.A. Topics in the theory of  $A_\alpha^p$  spaces. Leipzig: BSB Teubner, 1988.
2. Hedenmalm H., Korenblum B., Zhu K. Theory of Bergman spaces. Graduate Texts in Mathematics, 199. Springer-Verlag, New York, 2000.
3. Horowitz C. Zeros of functions in the Bergman spaces. Duke Math. J. – 1974, v. 41. № 4 – p. 698 – 710.

4. Беднаж В.А., Ермакова Д.С. Об одном свойстве корневых множеств функций  $A_\alpha^\infty$ , где  $\alpha > -1$ , на односвязной области комплексной плоскости // Современные проблемы теории функций и их приложения: материалы 21-й международной Саратовской зимней школы. Саратов.: Саратовский университет. 2022, с. 44-45.
5. Голузин Г.М. Геометрическая теория функций комплексного переменного. – М.: Наука, Гл. ред. физ.-мат. лит., 1966. – 630 с.
6. Джрбашян М.М. К проблеме представимости аналитических функций. Сообщ. ин-та матем. и мех. АН Арм. ССР, 1948.
7. Лаврентьев М.А., Шабат Б.В. Методы теории функций комплексного переменного. – М.: Наука, Гл. ред. физ.-мат. лит., 1958. – 736 с.

# Синхронизация волновых процессов в нейросетевых системах: возможности квантовых алгоритмов оценки фазы

*А.В. Боброва*

*Нижегородский государственный университет имени Н.И. Лобачевского*

Данная статья представляет собой исследование применения квантовых вычислений для задачи синхронизации волновых процессов в нейронных сетях. В статье описан процесс построения и тестирования квантового алгоритма оценки фазы (QPE). Были проведены эксперименты на простейшей модели нейронной сети с использованием квантовой цепи, разработанной на языке программирования Qiskit. Сравнение результатов с классическим методом корреляции показало, что квантовый алгоритм может быть использован для синхронизации волновых процессов в нейронных сетях, что может привести к повышению эффективности работы нейронных сетей в задачах обработки информации.

*Ключевые слова:* синхронизация волновых процессов, нейросетевые системы, квантовые алгоритмы, оценка фазы.

## 1. Введение

Синхронизация волновых процессов является одним из фундаментальных явлений, которые наблюдаются в различных областях физики, включая оптику, акустику, электронику, нейросетевые системы [1–4]. Однако при работе с большими объемами данных и сложными моделями нейросетей возникают проблемы с синхронизацией волновых процессов, которые могут привести к потере точности и ухудшению качества результата.

Синхронизация волновых процессов в нейросетевых системах может быть достигнута с помощью квантовых алгоритмов. В отличие от классических, квантовые методы могут обеспечить более точную и быструю синхронизацию, а также устойчивость к внешним помехам.

Для синхронизации волновых процессов в нейросетевых системах могут использоваться следующие квантовые алгоритмы:

1. Квантовая синхронизация Чурилова – Зимина: данный алгоритм использует пары связанных кубитов для синхронизации фаз двух волновых процессов в нейросетевых системах.
2. Алгоритмы на основе квантовых преобразований Фурье.
3. Квантовые алгоритмы на основе алгебраических методов: такие алгоритмы используются для решения задач синхронизации фаз в нейросетевых системах путем перевода задачи в задачу нахождения собственных значений квантовых операторов.

В данной работе рассмотрим квантовый алгоритм оценки фазы (QPE) и классический метод оценки фазы на основе корреляции.

## 2. Классический и квантовый алгоритмы оценки фазы

### 2.1. Классический алгоритм оценки фазы на основе корреляции

Классический алгоритм оценки фазы синхронизации волновых процессов в нейросетевых системах заключается в сравнении двух волновых форм. для определения степени сходства между ними. В случае синхронизации, это означает, что мы сравниваем две волны, чтобы определить, насколько близко они находятся в фазе.

Сначала выбирается опорная волновая форма (reference waveform), которая используется как эталон для сравнения. Затем для каждой другой волновой формы вычисляется корреляционная функция (correlation function), которая показывает, насколько сильно она совпадает с эталоном при различных задержках. Корреляционная функция вычисляется путем свертки (convolution) волновой формы с опорной волновой формой.

Для оценки фазы синхронизации используется максимальное значение корреляционной функции. Значение фазы определяется как задержка (delay) между максимальным значением и задержкой эталона (reference delay). Фаза синхронизации определяется как угол поворота фазы между волнами, который соответствует этому временному задержке.

Формально, корреляционная функция может быть записана следующим образом:

$$corr(t) = \int_{-\infty}^{\infty} f(\tau)g(t + \tau)d\tau,$$

где  $f(\tau)$  и  $g(\tau)$  – волновые формы для свертки. Значение  $corr(t)$  показывает степень совпадения между волнами  $f$  и  $g$  с задержкой  $t$ .

Значение фазы синхронизации может быть вычислено как  $2\pi(t_{max} - t_{ref})/T$ , где  $t_{max}$  – задержка с максимальным значением корреляционной функции,  $t_{ref}$  – задержка эталона,  $T$  – период волновой формы.

## 2.2. Квантовый алгоритм оценки фазы (QPE) [5]

Квантовый алгоритм оценки фазы синхронизации волновых процессов в нейросетевых системах (Quantum Phase Estimation, QPE) позволяет определять фазы собственных значений унитарных операторов. В данном случае, унитарный оператор, отображающий волновую функцию, будет иметь вид:

$$U = e^{-iHt},$$

где  $H$  – гамильтониан, определяющий эволюцию системы,  $t$  – время эволюции. Если мы применим оператор  $U$  к начальному состоянию  $|\psi_0\rangle$ , то получим состояние  $|\psi_t\rangle = U|\psi_0\rangle$ . Фаза синхронизации между двумя волновыми функциями будет определяться разностью между фазами соответствующих собственных значений оператора  $U$ .

Алгоритм QPE заключается в подготовке состояния, состоящего из двух регистров: регистра суперпозиции, содержащего  $n$  кубитов, и регистра собственного значения, содержащего  $m$  кубитов. Используется преобразование Уолша-Адамара для получения суперпозиции всех возможных состояний регистра суперпозиции. Затем применяется контролируемое преобразование  $U^{2^j}$ , где  $j$  – номер кубита регистра собственного значения,  $U$  – унитарный оператор. После этого на регистре собственного значения применяется обратное преобразование Фурье, а затем измерениями определяются фазы собственных значений.

Фаза синхронизации будет определяться как отношение целочисленной части результата QPE к общему количеству состояний регистра суперпозиции, умноженному на  $2\pi$ .

Более подробно, алгоритм может быть описан следующим образом:

1. Подготовить регистр суперпозиции  $|\psi\rangle$ , который состоит из  $n$  кубитов и находится в равномерной суперпозиции всех состояний:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle,$$

где  $N = 2^n$ . Каждому состоянию регистра суперпозиции соответствует угол  $\theta = \frac{2\pi k}{N}$ .

2. Применить оператор, отображающий функцию, к состоянию регистра суперпозиции. В данном случае, это оператор  $U$  отображает функцию нейронной сети  $f(x)$  на состояния регистра:

$$|\psi\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle |f(k)\rangle.$$

3. Применить алгоритм QPE для определения фазы синхронизации между волновыми функциями. Алгоритм QPE используется для оценки фазы оператора  $U$  в состоянии регистра суперпозиции. Алгоритм возвращает угол  $\theta'$ , который представляет собой приближенное значение фазы оператора  $U$ .

4. Фаза синхронизации между волновыми функциями вычисляется как:

$$\phi = \frac{\theta'}{2\pi},$$

где  $\phi$  – фаза синхронизации,  $\theta'$  – результат выполнения алгоритма QPE.

5. При необходимости, для повышения точности, можно повторить шаги 1–4 с увеличенным количеством кубитов в регистре суперпозиции или увеличенным числом шагов в алгоритме QPE.

### 3. Проведение эксперимента

Для проверки эффективности квантовой синхронизации в нейросетевых системах был проведен эксперимент. В эксперименте использовались инструменты квантовых вычислений, доступные в фреймворке Qiskit, такие как квантовые цепи и бэкенды для симуляции квантовых состояний.

Была создана квантовая цепь с двумя кубитами и двумя классическими регистрами. На кубиты применялись вращения по оси  $X$  и  $Z$ , а также контролируемые вентили NOT (CNOT). Параметры вращений были определены на основе модели нейронной сети, которую тестировали на синхронизации волновых процессов. Фрагмент квантовой схемы приведен на рис. 1.

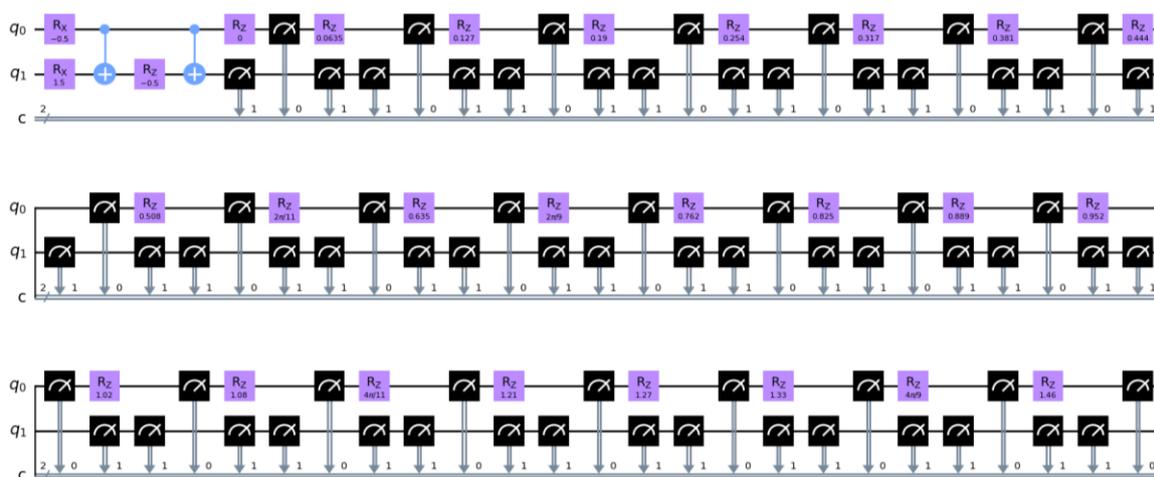


Рис. 1. Фрагмент квантовой схемы для оценки фазы при разных значениях  $\theta$ ,  $a = -0.5, b = 1.5, c = -0.5$

Для проведения эксперимента использовался симулятор квантовых вычислений, доступный в Qiskit. Он позволил получить результаты эксперимента на практике, без необходимости использования физических квантовых устройств. Было смоделировано 1024 испытания для каждого значения параметра  $\theta$ , чтобы получить достаточно точную статистическую выборку.  $\theta \in [0, 2\pi]$ .

Оценка точности квантового алгоритма QPE проводилась сравнением с классическим методом оценки фазы на основе корреляции.

В качестве нейросети была выбрана простейшая модель, представленная на рис. 2. Здесь  $x$  – входные данные,  $a, b, c$  – параметры нейронной сети.

```
def neural_network(x, a, b, c):
    return np.sin(x-a*b+ c) + np.cos(x-a*b+c)
```

Рис. 2. Модель нейросети

Результаты, полученные при разных значениях параметров  $a$ ,  $b$ ,  $c$ , представлены на рис. 3.

В ходе эксперимента были получены графики зависимости оценок фазы от значения параметра  $\theta$  для квантового и классического методов. Данные графики показывают, что квантовый метод дает более точные оценки фазы, чем классический метод, особенно при использовании малых значений параметра  $\theta$ . Кроме того, квантовый алгоритм имеет более высокую скорость работы, что позволяет решать задачу синхронизации при больших объемах данных.

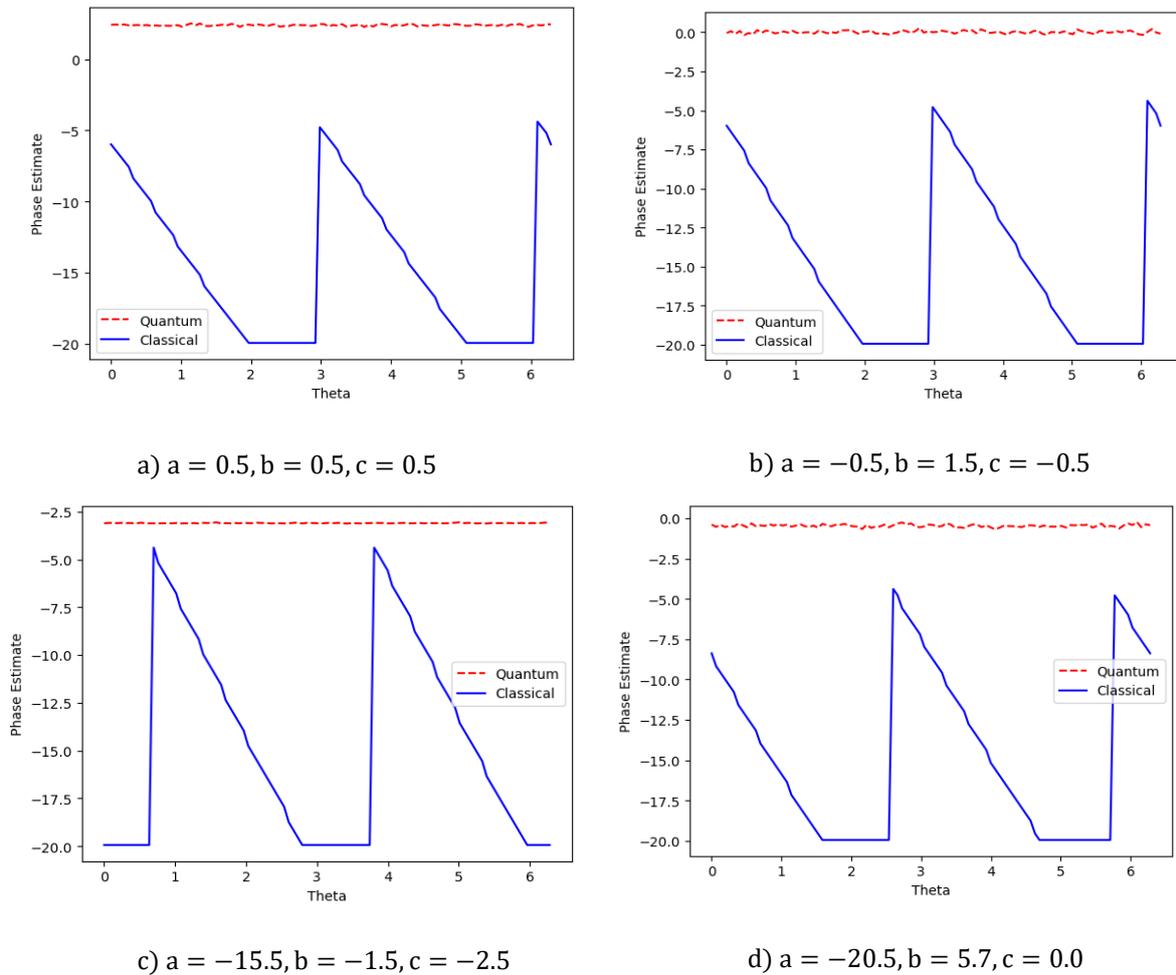


Рис. 3. Оценка фазы классическим и квантовым алгоритмами

## 4. Заключение

В данной статье рассмотрено применение квантовых вычислений для задачи синхронизации волновых процессов в нейросетевой системе. Описано использование квантового алгоритма оценки фазы (QPE), а также классического алгоритма оценки фазы на основе корреляции. Рассмотрено применение данных алгоритмов на простейшей нейросети. Доказано улучшение точности синхронизации алгоритмом QPE по сравнению с классическим методом оценки фазы на основе корреляции.

Продолжением данной работы может стать расширение метода на более сложные модели нейронных сетей, улучшение точности и стабильности синхронизации.

## Литература

1. Shepelev I.A., Bukh A.V., Strelkova G.I. Anti-phase synchronization of waves in a multiplex network of van der Pol oscillators. *Chaos, Solitons & Fractals*. 2022. Vol. 162.
2. Pikovsky, A., Rosenblum, M., Kurths, J. *Synchronization: A Universal Concept in Nonlinear Sciences* (Cambridge Nonlinear Science Series). Cambridge: Cambridge University Press. 2001. doi:10.1017/CBO9780511755743.
3. Arenas, A., Díaz-Guilera, A., Kurths, J., et al. Synchronization in complex networks. *Physics reports*. 2008.
4. Boccaletti, S., Kurths, J., Osipov, G., et al. The synchronization of chaotic systems. *Physics reports*. 2002.
5. Li, Xiantao. Some error analysis for the quantum phase estimation algorithms. *Journal of Physics A: Mathematical and Theoretical*. 2022.

# Улучшение стратегии управления рисками в системах с многоуровневой структурой

Д.С. Братухин

Нижегородский государственный технический университет им. Р.Е. Алексеева

Текст описывает проблемы многоуровневой структуры в больших корпорациях. Различные интересы на каждом уровне управления приводят к деформации целей и снижению эффективности структуры. Для улучшения эффективности нужно построить систему поощрений, где интересы участников проекта совпадают с интересами компании. Необходимо также создать систему стимулирования и разумное распределение рисков между участниками, зависящих от квалификации и вклада. В работе приводится модель взаимодействия между уровнями, которая может применяться в различных областях.

*Ключевые слова:* KPI, корпорации, многоуровневая структура, система поощрений, менеджер.

В крупных корпорациях, где присутствует многоуровневая структура, различные интересы в каждом уровне управления могут привести к деформации вектора целей и снижению эффективности структуры. Чтобы достичь максимальной эффективности, необходимо установить систему поощрений, в которой интересы всех участников проекта совпадают с интересами компании. Для успешного завершения проекта без возможности прямой оценки вклада каждого участника, следует создать систему стимулирования и справедливо распределить потенциальные риски между участниками, учитывая их квалификацию и вклад в проект.

В данной работе рассматривается модель взаимодействия только между двумя уровнями, так как это отражает особенности межуровневого взаимодействия внутри всей структуры. Считаем, что существует проект, успешность которого зависит от сложности, квалификации менеджера и его усилий. Целевая функция менеджера будет определяться величиной вознаграждения, учитывающей логарифмическую полезность и объем приложенных усилий (1), а целевая функция корпорации описывается уравнением (2):

$$F_m(E) = \frac{EQ \ln(M+1)}{D+EQ} - E, \quad (1)$$

$$F_c(M) = \frac{EQ(P-M)}{D+EQ}, \quad (2)$$

где  $P$  – возможный доход в случае успеха,  $D$  – сложность проекта,  $M$  – награда, которую собственник определяет для менеджера в случае его успеха,  $Q$  – квалификация менеджера и  $E$  – объем усилий, который менеджер приложит к проекту.

Для того, чтобы правильно решить задачу на первом этапе проекта, менеджер должен определить необходимое количество ресурсов для достижения нужного результата. Для этого важно также определить объем усилий, который потребует менеджеру для достижения цели при рациональном подходе (3).

$$E_{optM} = \operatorname{argmax}_{E \geq 0} \frac{EQ \ln(M+1)}{D+EQ} - E. \quad (3)$$

Учитывая предполагаемую реакцию менеджера на систему поощрения, следует определить его вознаграждение в следующей форме (4).

$$M_{optC} = \operatorname{argmax}_{R \geq 0} \frac{E_{optM} Q (P-M)}{D + E_{optM} Q}. \quad (4)$$

В результате принятия оптимальных решений собственника и менеджера, целевая функция корпорации будет иметь следующий вид (5).

$$F_c(M) = \frac{E_{optM} Q (P - M_{optC})}{D + E_{optM} Q} \quad (5)$$

Важно учитывать ограничения на вознаграждение (6) при решении задачи. Если левая и правая части неравенства (3) будут равны, то целевые функции будут равны нулю. Учитывая условие (6), задача оптимизации вознаграждения (5) принимает вид (7) после упрощения.

$$M \geq \exp\left(\frac{D}{Q}\right) - 1 \quad (6)$$

$$M_{optC} = \arg \max_{M \geq \exp\left(\frac{D}{Q}\right) - 1} \left[ (P - M) \left\{ 1 - \frac{1}{\sqrt{\frac{Q}{D} \ln(M + 1)}} \right\} \right] \quad (7)$$

Вместо проблемного решения алгебраического уравнения с дробями и логарифмами, предлагается применить численное решение задачи максимизации (7) по переменной  $M$ .

Предлагается численное решение задачи максимизации (7) по переменной  $M$ , вместо того, чтобы решать сложное алгебраическое уравнение с дробями и логарифмами. Для оптимального решения задачи необходимо учитывать ограничения на вознаграждение (6). При равенстве левой и правой части неравенства (3), целевые функции будут равны нулю.

По результатам анализа можно определить оптимальные решения, которые могут быть использованы как корпорацией, так и менеджером, с учетом параметров  $Q = 1$ ,  $D$  и варьируемых переменных  $P$ . Оценка проектов будет зависеть от дохода, сложности и затрат, а вознаграждение для менеджера будет связано со сложностью проекта и прогнозируемым доходом, а также с увеличением дохода от проекта оптимальная доля вознаграждения уменьшается.

Важность полученных результатов заключается в представленной модели, которая помогает находить баланс между интересами корпорации и менеджеров на всех уровнях. Для менеджера крайне важно осознанно оценить последствия своих действий при достижении успеха проекта, управлении рисками и повышении прибыли. Эта модель простой мотивации может использоваться в ситуации, когда у менеджера есть только один проект. Она также может служить прототипом поощрений для менеджеров при проектах с риском неудачи. Данный подход может быть адаптирован для многоуровневых корпоративных систем управления.

## Литература

1. Масленников Д. А., Катаева Л. Ю. Модель двухуровневой оптимизации управления для обеспечения экологической безопасности. // Развитие и безопасность. 2022. № 3. С. 103–114.
2. Катаева Л. Ю., Масленников Д. А. Многокритериальный подход к обеспечению экологической безопасности. // Развитие и безопасность. 2021. № 3. С. 96–104.

# Оптимизация аэродинамического профиля. Крыло с изменяемым профилем

*Р.М. Вольников*

*Нижегородский государственный технический университет им. Р.Е. Алексеева*

При проектировании БПЛА важной частью является оптимальный профиль крыла с наименьшим сопротивлением. Именно от этой характеристики зависит оптимальность, рациональность и КПД летательного средства. Один из способов снижения сопротивления – сведение к минимуму сопротивления трения на поверхности крыла. В данной статье решается задача поиска оптимального профиля крыла для режима барражирования беспилотника.

*Ключевые слова:* БПЛА, профиль крыла, оптимальный свободный полёт, аэродинамика, барражирование, моделирование, аэродинамика.

Технология беспилотных летательных аппаратов, которая возникла в военных целях, в настоящее время широко используется в коммерческих, профессиональных, промышленных и частных целях. Беспилотные летательные аппараты (БПЛА), широко известные как «дроны», охватывают различные отрасли экономики, например, сельское хозяйство, транспорт, инфраструктуру, развлечения и телекоммуникации. Дроны – это не только экологически чистые гаджеты, которые позволяют сократить количество выбросов углекислого газа, но и экономят время и деньги. Таким образом, дроны могут оказаться важной силой во благо, поскольку они обладают огромным потенциалом для использования для достижения целей устойчивого развития (ЦУР), установленных Организацией Объединенных Наций и принятых в 2015 году. В современной картине политической ситуации, сложившейся в мире, актуальность внедрения новых технологий, в том числе и БПЛА, всё больше набирает обороты.

Главной целью работы является расчёт оптимального профиля крыла беспилотника для режима барражирования. Барражирование – такой полёт, при котором летающий аппарат способен обладать максимальной длительностью полёта. В данном вопросе важна аэродинамическая часть. Ключевой составляющей в такой задаче является оптимальный профиль крыла. Один из вариантов повышения КПД дрона – это достижение уменьшения сопротивления крыла. Для решения этой задачи рассмотрены существующие технологии и образцы БПЛА. Так, например, в данный момент на вооружении в Российской Федерации используется «Орлан-10». Это беспилотный комплекс, обладающий такими характеристиками, как протяжённость полёта до 16 часов и дальностью до 600 км. Данный экземпляр применялся в гражданской войне в Сирии, а также с 2014 года в вооружённом конфликте с Украиной.



**Рис. 1.** Получение точек для базовой геометрии профиля крыла при помощи Airfoil plotter

В данной работе выбран прототип профиля НАСА и произведена его модернизация для улучшения аэродинамических и лётных характеристик. Сперва были получены точки для базовой геометрии крыла при помощи генератора точек Airfoil.

Далее была решена задача по поиску оптимального числа Рейнольдса. Путём смещения точки перегиба на верхней поверхности профиля крыла (рис. 5) можно улучшить аэродинамические качества. Достигается это поиском оптимального соотношения фаз турбулентного и ламинарного движений.

Сперва было необходимо создать симуляцию потока, распределение давления и скорости. Для этого была использован пакет OpenFoam с blockMesh. Данное ПО работает под ОС Linux. Сложная форма профиля аппроксимируется сплайном. На рисунке 2 изображено разделение на ключевые блоки, в которых воздушный поток меняет своё направление. Это общепринятый подход во всём мире. Точки 1 и 16, а также 5 и 17 имеют одни и те же координаты. Однако, поскольку они определены отдельными точками, блоки 2 и 4 не имеют общих граней. Поэтому для соединения этих 2-х блоков используется функция mergePatch. Пограничный слой на аэродинамическом профиле обеспечивается сложной градацией краев, поэтому нет необходимости выделять пограничный слой в другой блок.

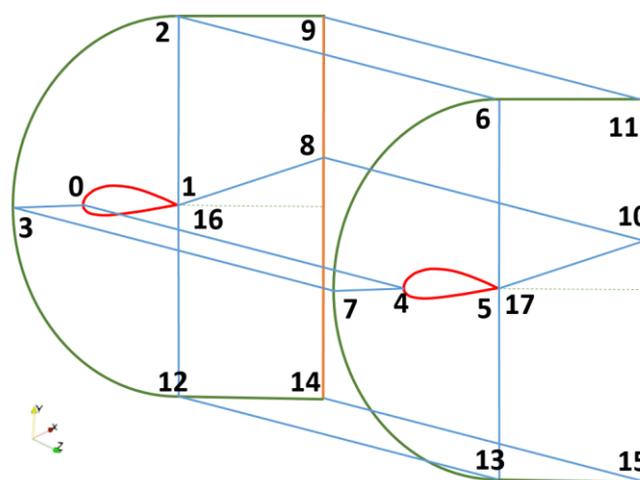


Рис. 2. Деление на блоки и ключевые точки в OpenFoam

С помощью пакета OpenFoam был смоделирован воздушный поток, в результате которого экспериментальным путём был получен оптимальный коэффициент Рейнольдса. В данном ПО есть функция по поиску оптимального решения для заданного параметра.

На рисунке 3 смоделировано распределение давления по поверхности крыла в OpenFoam.

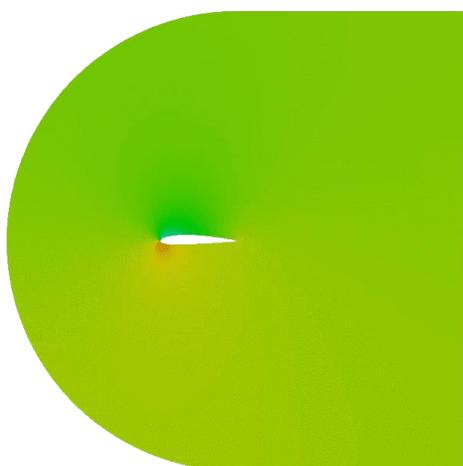


Рис. 3. Распределение давления

На рисунке 4 смоделировано распределение скоростного потока по поверхности крыла в OpenFoam.

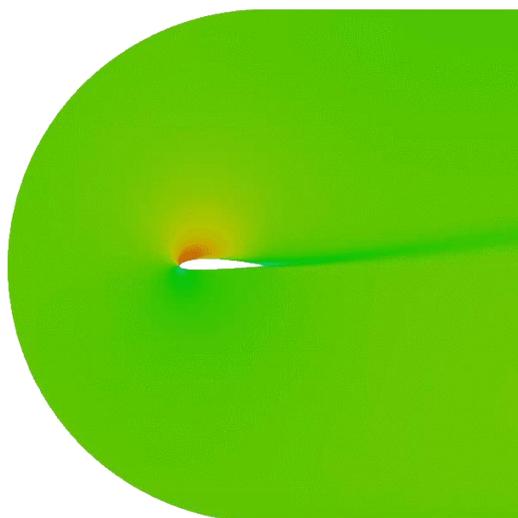


Рис. 4. Распределение скоростного потока

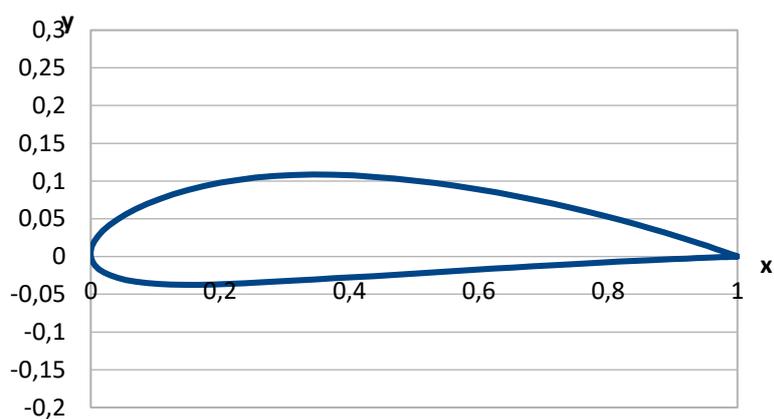


Рис. 5. Геометрия модернизированного профиля крыла в Microsoft Excel

На рисунке 6 показаны линии потока.

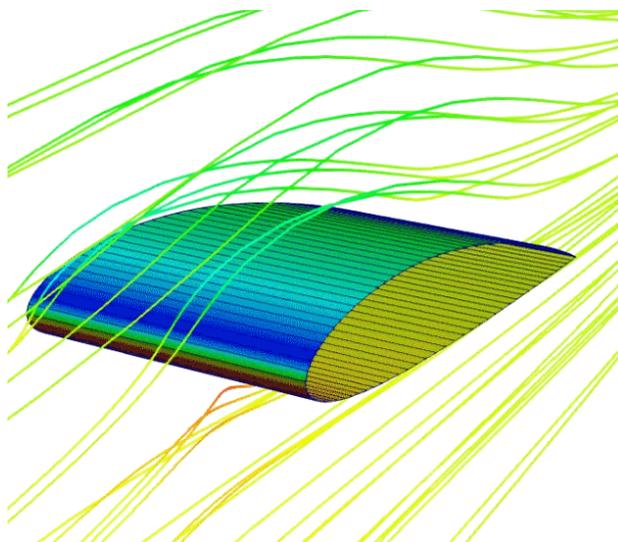


Рис. 6. Симуляция линий потока воздуха

Данный пакет автоматически подсчитывает подъёмную силу и сопротивление воздуха. Таким образом, оптимальным смещением точки перегиба на 5% ближе к хвосту крыла даёт прирост коэффициента подъёмной силы  $\Delta C_{ya} = 0,0122$  и уменьшение коэффициента аэродинамического сопротивления  $\Delta C_{xa} = 0,000334$ .

## Литература

1. C. Ostowari, D. Naik - "Post-stall wind tunnel data for NACA 44XX series airfoil sections: a subcontract report", Final Report prepared under SERI Task No. 4807.20.
2. Eastman N. Jacobs, William C. Clay - NASA Report No. 530 "Characteristics of the N.A.C.A. 23012 airfoil from tests in the full-scale and variable-density tunnels".
3. Катаева Л. Ю., Масленников Д. А., Михалев С. В., Киселева Н. Н. Поиск оптимальных режимов полета тела на основе аппроксимации результатов CFD-моделирования // Труды НГТУ им. Р.Е. Алексеева. 2019. № 4 (127). С. 27–35. EDN: АНРНМ.
4. Численное решение задач экономики с использованием EXCEL, C++ и MATLAB: учебное пособие / Л. Ю. Катаева, М. Н. Ильичева, Т. А. Федосеева, Д. А. Масленников. – Н. Новгород: Нижегородский государственный технический университет им. Р. Е. Алексеева, 2020. – 229 с.
5. Численные методы решения прикладных задач: учеб. пособие / Л. Ю. Катаева, Д. А. Масленников, Н. А. Лощилова [и др.]. – Н. Новгород: Нижегородский государственный технический университет им. Р. Е. Алексеева, 2014. 283 с.

# Автоматизация подготовки рецепта на медицинские изделия

*А.Т. Герюгова*

*Карачаево-Черкесский Государственный университет имени У. Д. Алиева*

В данной работе предлагается вариант проектного решения автоматизации подготовки рецепта на медицинские изделия. Данное решение позволит ускорить процесс создания рецепта, облегчить работу фармацевту, а также позволит вести учёт данных пациентов и выписанных ранее рецептов. Рассматривается процесс проектирования приложения, создания баз данных, реализация интерфейса пользователя и принцип работы самой программы.

*Ключевые слова:* автоматизированная информационная система, база данных, приложение, Delphi, рецепт на медицинское изделие.

## Введение

Рецепт на медицинские изделия – это письменное обращение врача к фармацевту о приготовлении и отпуске лекарств. Характеристикой, по которой препараты разделяются на рецептурные и безрецептурные, является потенциальная угроза злоупотребления.

Рецепт является важнейшим врачебным документом, от правильного выбора лекарственного средства, его дозы, рациональной комбинации лекарственных веществ во многом зависит успех лечения больного. Врач несет полную юридическую ответственность за выписанный рецепт. Эта особенность накладывает чрезвычайно строгие и серьезные требования к лекарству по качеству, терапевтической эффективности, чистоте, стабильности, точности дозировки. Назначение лекарственных препаратов для медицинского применения осуществляется лечащим врачом, фельдшером, акушеркой в случае возложения на них полномочий лечащего врача, при оказании медицинской помощи в медицинских организациях [6].

Автоматизация данного процесса может ускорить процесс создания рецепта, облегчить работу фармацевту, а также позволит вести учет данных пациентов и выписанных ранее рецептов.

Медицинский работник, назначивший лекарственный препарат, оформляет назначение лекарственных препаратов, оформляет назначение лекарственных препаратов, в том числе подлежащих изготовлению и отпуску аптечными организациями, на рецептурном бланке, оформленном на бумажном носителе за своей подписью, и (или) с согласия пациента или его законного представителя на рецептурном бланке в форме электронного документа.

В рецепте выделяют 10 частей [6]:

1. Штамп лечебно-профилактического учреждения и его шифр. В штампе должны быть указаны: название МО, адрес МО, номер телефона. Необходимы для установления связи аптеки с лечебным учреждением и решения различных вопросов.

2. Серия рецептурного бланка.

3. Номер рецептурного бланка.

4. Дата выписывания рецепта. С этого момента отсчитывается срок действия рецепта.

5. ФИО пациента, для которого предназначен лекарственный препарат.

6. Дата рождения пациента, для которого предназначен препарат.

7. Адрес или номер медицинской карты пациента, для которого предназначен препарат.

8. ФИО врача. Медицинские работники выписывают рецепты за своей подписью.

9. Должность врача.

10. Подпись врача.

Запрещается выписывать рецепты на лекарственные препараты в следующих случаях [6].

## 1. Реализация приложения

Диаграмма на рисунке 1 отражает структуру программного приложения, которое будет состоять из 5 модулей. Главный модуль – “Create a prescription” – будет реализовывать состояние «отображение рецепта», а также будет содержать вызовы вспомогательных модулей.

Первый дополнительный модуль – “Enter doctor’s data” – реализует состояние «ввод данных врача». С помощью модуля “Enter patient data” осуществляется состояние «ввод данных пациента». Модули “Enter medication data” и “Keep a record of recipes” реализуют состояния «ввод данных медицинского изделия» и «ведение журнала рецептов».

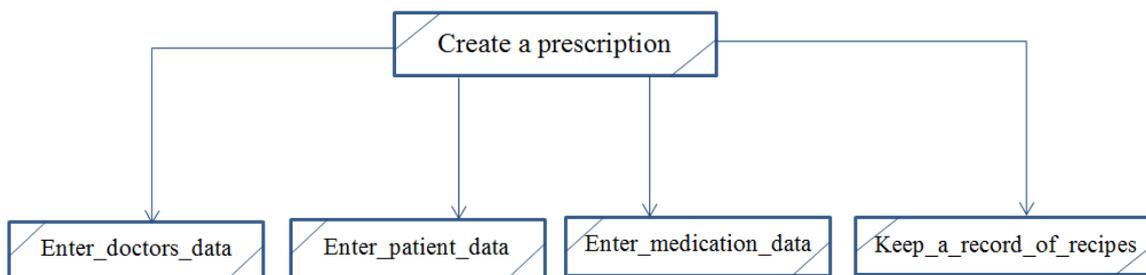


Рис. 1. Диаграмма структуры программного приложения

Диаграмма потоков данных (рис. 2) отражает передачу информации от одной функции к другой. Внешней сущностью системы подготовки рецепта на медицинские изделия является врач, который вводит данные пациента и медицинского изделия, собственные предписания, и по запросу получает готовую версию рецепта, а также может просмотреть введенные ранее данные.

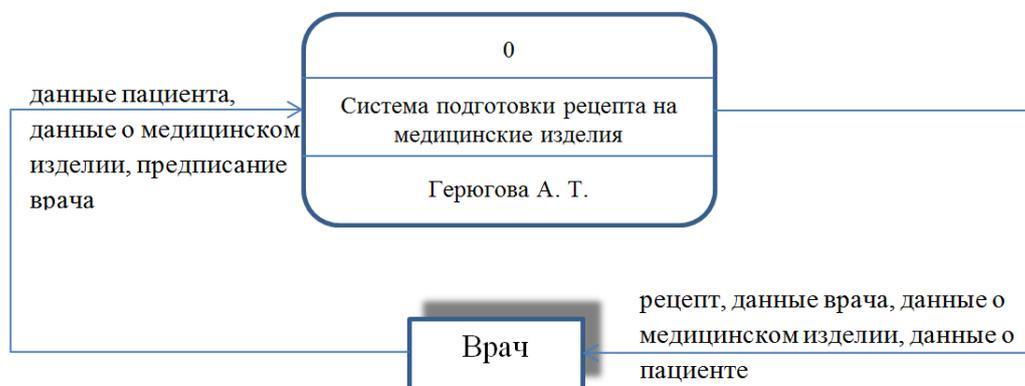


Рис. 2. Контекстная диаграмма системы подготовки рецепта на медицинские изделия

На рисунке 3 изображена диаграмма переходов состояний системы.

Система может находиться в четырех различных состояниях – «Отображение рецепта», «Ввод данных медицинского изделия», «Ввод данных пациента», «Ввод данных врача» и «Ведение журнала рецептов». Начальным состоянием является «Отображение рецепта».

Для перехода из начального к состоянию «Ввод данных медицинского изделия» используется событие «медицинское изделие». Также можно использовать событие «ок», чтобы добавить данные в базу данных, и событие «отмена», чтобы отменить предыдущий шаг.

Для перехода к состоянию «ввод данных пациента» используется состояние «пациент». Аналогично, для переходов к состояниям «ввод данных врача» и «ведение журнала рецептов» используются состояния «врач» и «журнал» соответственно.

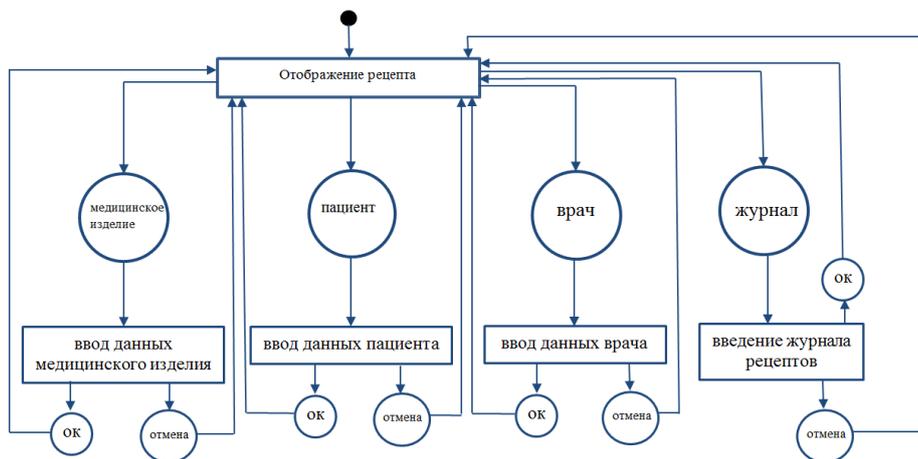


Рис. 3. Диаграмма переходов состояний системы подготовки рецепта на медицинские изделия

Для реализации базы данных использовалось приложение «Microsoft Access» (рис. 4). В таблице «Врач» ключевым полем является поле «код врача», которое обеспечивает уникальность записей и реализацию связи. В таблице «Пациент» ключевое поле – «номер медицинской карты»; в таблице «Рецепт» используются два ключевых поля – «серия» и «номер» (рецепта). Также в таблицу «Рецепт» добавлены два внешних ключевых поля – «код врача» и «номер медицинской карты». Данные ключевые поля будут обеспечивать целостность связи таблиц «Рецепт», «Врач» и «Пациент».

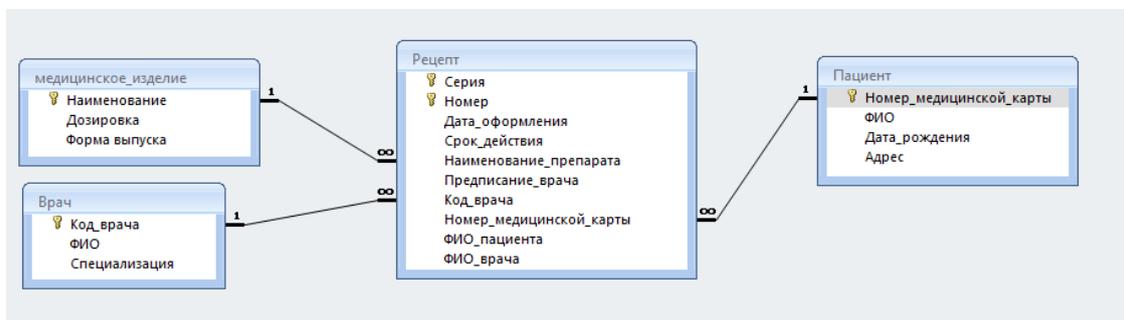


Рис. 4. Схема таблиц

Рис. 5. Главное окно программы

Для реализации функций системы подготовки рецепта на медицинские изделия потребуется 5 окон. Основное окно показано на рисунке 5, оно содержит средства ввода данных о рецепте и кнопки для открытия дополнительных окон, а также кнопку для печати рецепта. С помощью дополнительных окон пользователь может ввести все необходимые данные в базу данных, а именно – данные врача, данные пациента и информацию о медицинском изделии. В дополнительном окне «Журнал» будет вестись учёт всех выпущенных ранее рецептов.

При запуске программы открывается главное окно приложения (рис. 5), в котором пользователь может заполнить шаблон рецепта либо открыть дополнительные окна.

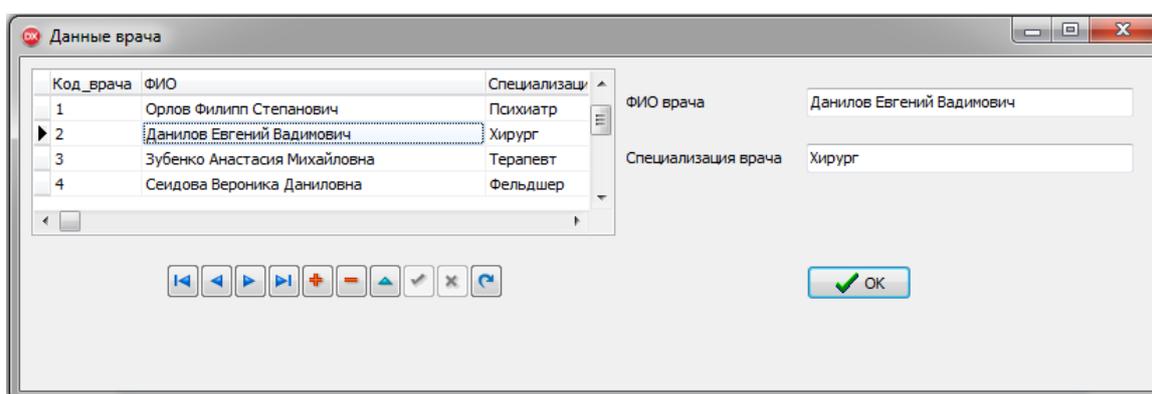


Рис. 6 (а). Окно ввода данных врача

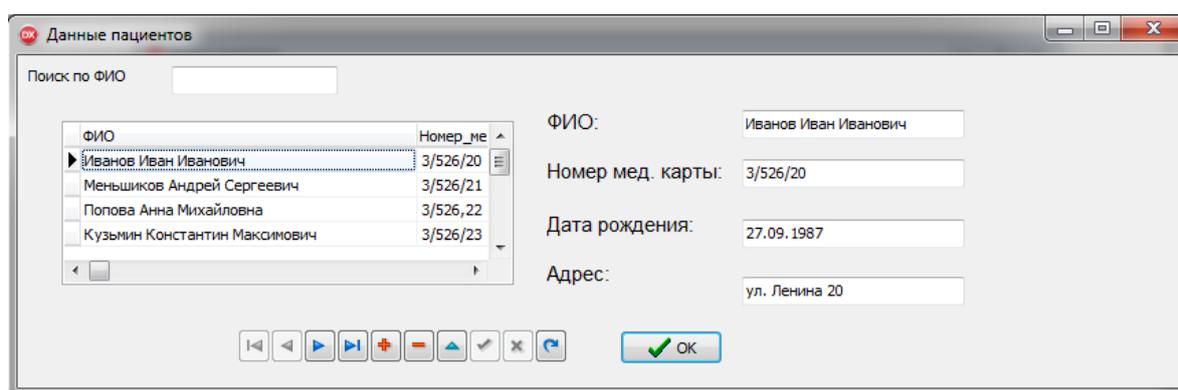


Рис. 6. (б). Окно ввода данных пациента

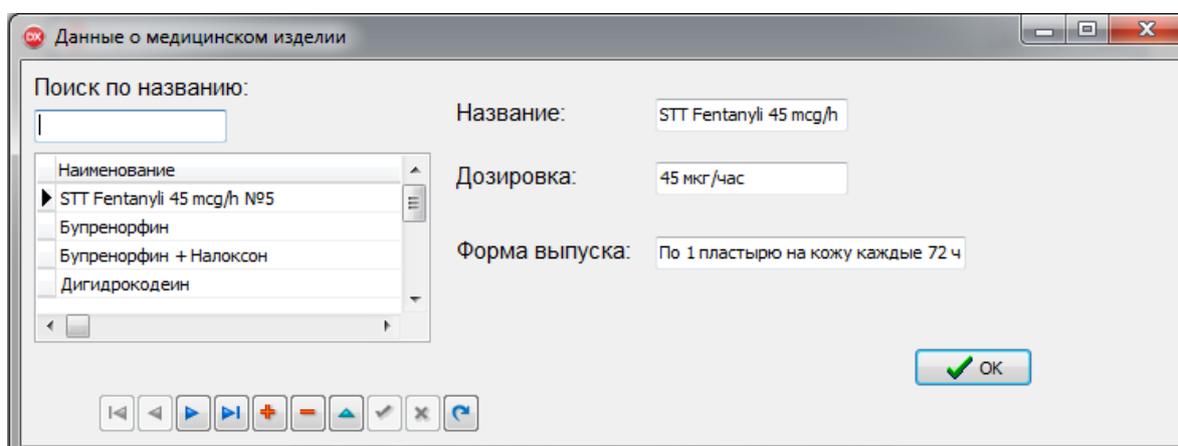


Рис. 6 (в). Окно ввода данных о медицинском изделии

Для доступа к базе данных используется технология ADO (ActiveX Data Objects). Основные объекты, используемые в разработке – TADO Table и TADOQuery. Ввод и редактирование текста выполняются в полях Edit и DBEdit. Просмотр записей из базы данных осуществляется с помощью компонента DBGrid. Компонент DBComboBox отображает данные из базы данных в виде комбинированного списка. Также на форму добавлены компоненты Button – 6 кнопок. Кнопки «Врач», «Пациенты», «Медицинское изделие» открывают дополнительные окна для ввода данных (рисунки 6 (а), 6 (б), 6 (в)).

На всех трёх окнах присутствуют компоненты DBGrid – таблица, отображающая информацию из базы данных, и DBEdit – связанное с данными окно редактирования, а также компонент DBNavigator – совокупность командных кнопок, обеспечивающих указателя текущей записи, активизация режима редактирования добавления и удаления записей [4].

Кнопка «Печать» с помощью модуля ComObj и функций CreateOleObject, FindAndInsert открывает заранее созданный, единичный шаблон текстового документа, заменяя текст внутри метки ###<...>& на данные, введенные пользователем в строки редактирования Edit и DBEdit.

Кнопка «Добавить» с помощью компонента TADOQuery и команды «Insert into Рецепт» добавляет в таблицу новые значения (values).

Кнопка «Журнал» открывает окно «Журнал», в котором с помощью компонентов DBGrid и DBEdit выводятся данные о созданных ранее рецептах.

Все поясняющие надписи реализованы компонентом Label.

## 2. Тестирование системы

После запуска программы открывается главное окно, на котором расположен шаблон рецепта и все вспомогательные кнопки. Поля «Фамилия, инициалы пациента», «Фамилия, инициалы лечащего врача», «Предписание врача» могут заполняться с помощью выпадающего списка с функцией поиска. Поля «Дата рождения», «Адрес места жительства или № медицинской карты», «Дозировка» и «Форма выпуска» привязаны к перечисленным выше полям и заполняются автоматически. В поле «дата оформления» с момента запуска вводится сегодняшняя дата. Все остальные поля заполняются пользователем (врачом). Таким образом, шаблон рецепта заполнен. Далее пользователь может переслать все заполненные данные на текстовый шаблон рецепта и отправить на печать.

Код формы по ОКУД 3108805  
Медицинская документация  
Форма № 148-1/у-88  
Утверждено приказом  
Министерства здравоохранения  
Российской Федерации от  
14 января 2019 г. № 4 н.

**Рецепт** серия 49 РЛ номер 991594  
27.12.2022  
дата оформления  
(взрослый, детский)

Фамилия, имя и отчество пациента Меньшиков  
Андрей Сергеевич

Дата рождения 15.01.1994

Адрес места жительства или № медицинской карты  
3/526/21

Фамилия, имя и отчество лечащего врача  
Орлов Филипп Степанович

Рр: Бупренорфин+ Налоксон 0,2 мг + 0,2 мг таблетки  
сублингвальные

Рис. 7. Заполненный текстовый шаблон

При нажатии на кнопку «Добавить» все введенные данные будут добавлены в базу данных (будет создана новая запись в таблице «Рецепт»), и в журнал, в одноименном окне. Все добавленные ранее рецепты будут отображаться в окне «Журнал», открыть которое можно нажав на кнопку «Журнал» (рис. 19).

Кнопки «Врач», «Пациент», «медицинское изделие» открывают окна. В каждом из данных окон могут просматриваться, редактироваться, добавляться и удаляться данные о враче, пациенте и препарате соответственно.

## Заключение

Система подготовки рецепта на медицинское изделие была разбита на 4 функциональные подсистемы:

- 1) Подсистема ввода и редактирования данных о враче;
- 2) Подсистема ввода и редактирования данных о медицинском изделии;
- 3) Подсистема ввода и редактирования данных о пациенте;
- 4) Введение учета и создание рецепта.

Главное окно системы содержит шаблон рецепта и кнопки для переключения на дополнительные окна. В дополнительных окнах находятся окна редактирования, в которые могут вводиться данные.

Хранение данных системы подготовки рецепта на медицинские изделия реализуется следующими таблицами:

«Рецепт» является основной таблицей и с помощью связи «один-ко-многим» может получать данные от связанных таблиц, а также обеспечивать целостность данных. В данной таблице находятся все данные о рецепте;

Таблица «Врач» содержит в себе все необходимые данные о враче, а именно код врача, ФИО и специализацию;

В таблице «Пациент» находятся данные пациента: номер медицинской карты, ФИО, дата рождения и адрес;

Таблица «медицинское изделие» содержит данные о препарате: наименование, дозировка и форма выпуска.

Система подготовки рецепта на медицинское изделие реализована с помощью технологии ActiveX Data Objects, которая позволяет получить доступ к базе данных, и технологии Com, которая экспортирует данные из приложения во внешний файл формата doc.

Тестирование показало, что спроектированная программа выполняет все заданные функции, включая ввод данных в базу данных и печать рецепта.

**Благодарности:** Выражаю благодарность моему научному руководителю А.М. Узденовой, к. ф.-м. н., доценту кафедры информатики и вычислительной математики КЧГУ за значимые обсуждения и замечания.

## Литература

1. Дарахвелидзе П.Г., Марков Е.П. Delphi 2005 для Win32. СПб.: БХВ-Петербург, 2005. 1136 с.
2. Диго С.М. Базы данных. Проектирование и создание: учебно-методический комплекс. Москва: ЕАОИ, 2008.
3. Классификация информационных технологий // Интуит. Национальный открытый университет [сайт]. URL: <https://intuit.ru/studies/courses/3609/851/lecture/31646?page=4>
4. Компоненты: DBNavigator // Delphi — краткий справочник [сайт]. — URL: <https://programm.ws/page.php?id=760>
5. Прокофьева Л.В. Общая рецептура: учебно-методическое пособие / Л.В. Прокофьева, Е.Ю. Насырова, Е.С. Долгова, А.Е. Кармышева. Ульяновск: УГУ, 2017. С. 9–12.

6. Симонова И.В. Общая рецептура: учеб. пособие / И. В. Симонова, Р. А. Анохина. Благовещенск: АГМА, 2022. С. 14–19.
7. Смирнова Г.Н. Проектирование экономических информационных систем: учебник // Г.Н. Смирнова, А.А. Сорокин, Ю.Ф. Тельнов. Москва, 2002.
8. Узденова А.М. Проектирование информационных систем. Часть 1: учебно-методическое пособие. КЧГУ. Карачаевск: Изд-во КЧГУ, 2015. С. 89–99.

# Приложение, рассчитывающее абсолютные и относительные показатели качества программного средства на основе иерархической модели, приведенной в ГОСТ 28195-99

*Г.И. Захарычев*

*Саровский физико-технический институт СарФТИ НИЯУ МИФИ*

Работа состоит из описания программы, созданной в среде программирования QtCreator 5.12.1. Программа представляет собой приложение, которое на основе иерархической модели качества программных средств, приведенной в стандарте ГОСТ 28195-99, приводит оценки качества программного средства и рассчитывает абсолютные и относительные показатели различных уровней.

*Ключевые слова:* программное средство, иерархическая модель качества программного средства, стандарт, гост, оценка качества, программная инженерия

## 1. Введение

В настоящее время люди во всех сферах почти своей жизнедеятельности используют компьютеры. От правильности работы программных средств, используемых компьютерами, во многом зависят результаты труда и безопасность тысяч людей. Из-за этого люди все больше и больше стали уделять внимание стандартизации программных средств (ПС) и процессов их разработки, прогнозированию, оценке и управлению качеством программных средств на протяжении всего их жизненного цикла.

Многие компании разрабатывают собственное ПО, предназначенное для оценки и управления качеством ПС. Но большинство этих программ либо коммерческие, либо имеют закрытый код, что делает доступ к ним широкого пользователя проблематичным.

Моя разработка, имеет открытый код и позволяет наглядно оценить качество ПС, используя факторы качества, описанные в ГОСТ 28195-99. Среда ее применения – оценка качества прикладных программ для управления техническими устройствами и технологическими процессами.

## 2. Анализ задания, выбор технологии, языка и среды разработки

Для разработки программы оценки качества ПС, были поставлены следующие задачи:

1) Изучить разделы стандарта ГОСТ 28195. В теоретической части работы привести краткое описание стандарта, его общее назначение и отдельных разделов.

2) Провести сравнительный анализ стандартов ГОСТ 28195, ИСО 9126, ГОСТ 28806. Сделать выводы.

3) На основе иерархической модели оценки качества ПС, приведенной в стандарте, разработать программу оценки качества программного средства и расчета абсолютных и относительных показателей качества различных уровней. Эталонные значения показателей качества различных уровней взять из таблиц стандарта ГОСТ 28195-99, выборка произвольная. Значения показателей качества оцениваемого ПС сгенерировать случайным образом. Сравнительный анализ показателей качества представить в табличной форме.

4) Программа разрабатывается на языке C++, стиль ООП, наличие собственных классов, среда разработки любая, оконное/диалоговое приложение, Windows-интерфейс.

5) Оценить возможность использования разработанной программы с учетом требований прочих стандартов оценки качества ПС (ИСО 9126, ГОСТ 28806). Сделать выводы.

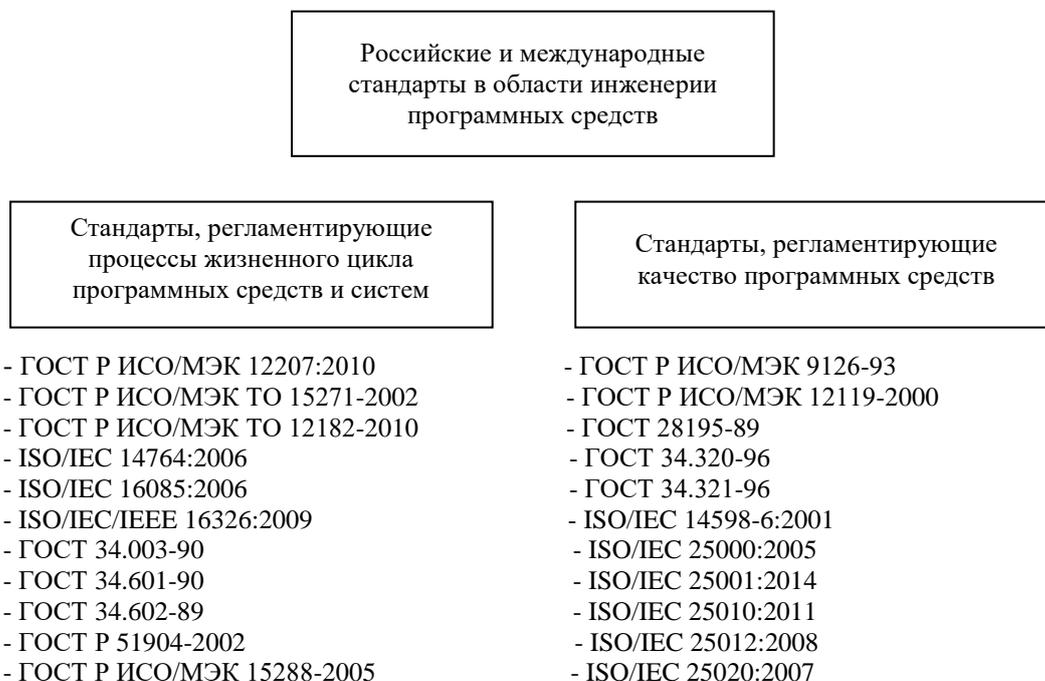
6) Оформить пояснительную записку согласно требованиям.

Для реализации моего проекта, я выбрал язык C++ и среду программирования Qt 5.12.1, так как она позволяет реализовывать ООП, а также в наглядном виде представлять результаты расчетов.

### 3. Теоретическая часть

#### 3.1. Российские и международные стандарты в области инженерии программных средств

Разработка информационных систем в последние годы является распространённой и важной задачей. Очевидно, что программные инженеры должны быть способными разрабатывать ПС с помощью наилучших практических решений с долговременной перспективой.



**Рис.1.** Стандарты в области инженерии программных средств

Наиболее адекватной представляется разработка архитектуры сложных прикладных информационных систем на основе эффективного объединения разных видов оборудования и программного обеспечения, применения стандартизованных интерфейсов между компонентами системы и т.п.

Дальнейшее рассмотрение будет посвящено нескольким основополагающим стандартам в области инженерии программных средств, охватывающим процессы жизненного цикла программных средств и систем и качество ПС (рис. 1).

#### 3.2. ГОСТ Р ИСО/МЭК 9126-93 «Характеристики качества и руководства по их применению»

Настоящий стандарт определяет шесть характеристик, которые с минимальным дублированием описывают качество программного обеспечения. Данные характеристики образуют основу для дальнейшего уточнения и описания качества программного обеспечения. Руководства описывают использование характеристик качества для оценки качества программного обеспечения. Настоящий стандарт не определяет подхарактеристики (комплексные показатели) и показатели, а также методы измерения, ранжирования и оценки. Данный стандарт придерживается определения качества по ИСО 8402. Определения характеристик и соответствующая модель процесса оценки качества, приведенные в настоящем стандарте, применимы тогда, когда определены требования для программной продукции и оценивается её качество в процессе жизненного цикла. Эти характеристики могут применяться к любому виду программного обеспечения, включая программы ЭВМ и данные, входящие в

программно-технические средства (встроенные программы). Настоящий стандарт предназначен для характеристик, связанных с приобретением, разработкой, эксплуатацией, поддержкой, сопровождением или проверкой программного обеспечения.

Качество программного обеспечения может быть оценено следующими характеристиками:

1. Функциональные возможности (Functionality).
2. Надежность (Reliability).
3. Практичность (Usability).
4. Эффективность (Efficiencies).
5. Сопровождаемость (Maintainability).
6. Мобильность (Portability).

### **3.3. ГОСТ 28195-89 «Оценка качества программных средств. Общие положения»**

Настоящий стандарт устанавливает общие положения по оценке качества ПС вычислительной техники, номенклатуру и применяемость показателей качества в зависимости от назначения и области применения. ГОСТ 28195–99 определяет оценку качества программного средства как совокупность операций, включающих выбор номенклатуры показателей качества оцениваемого программного средства, определение значений этих показателей и сравнение их с базовыми значениями.

В соответствии с данным стандартом оценка качества должна проводиться применительно ко всем работам жизненного цикла ПС при планировании показателей качества ПС, контроле качества в процессе разработки, проверке эффективности модификации ПС в процессе сопровождения. Стандартом ГОСТ 28195–99 рекомендован метод оценки обобщенного качества программного обеспечения, основанный на иерархической модели качества.

ГОСТ 28195–99 предлагает следующую терминологию для показателей качества каждого уровня:

- уровень 1 – факторы качества;
- уровень 2 – критерии качества;
- уровень 3 – метрики;
- уровень 4 – оценочные элементы или единичные показатели.

### **3.4. Описание ГОСТ 28195-99**

ГОСТ 28195-99 регламентирует выполнение оценки качества программных средств и систем на основе иерархической модели качества. В соответствии с данной моделью совокупность свойств, отражающих качество программного средства, представляется в виде многоуровневой структуры.

ГОСТ 28195-99 определяет четырёхуровневую иерархическую модель оценки качества ПС.

На первом уровне модели находятся факторы качества (соответствуют характеристикам качества), на втором уровне – критерии качества (подхарактеристики качества), на третьем – метрики, на четвертом – оценочные элементы.

Номенклатура факторов и критериев является обязательной, а номенклатура метрик и оценочных элементов – рекомендуемой.

Факторы оценки:

- 1) Надежность;
- 2) Сопровождаемость;
- 3) Удобство использования;
- 4) Эффективность;
- 5) Универсальность;
- 6) Функциональность.

Для каждого из выбранных факторов качества составляется четырёхуровневая иерархическая модель, отражающая взаимосвязь факторов, критериев, метрик и оценочных элементов.

Критерии оценки:

- 1.1) Устойчивость функционирования;

- 1.2) Работоспособность;
- 2.1) Структурность;
- 2.2) Простота конструкции;
- 2.3) Наглядность;
- 2.4) Повторяемость;
- 2.5) Полнота документации;
- 3.1) Легкость освоения;
- 3.2) Доступность программных документов;
- 3.3) Удобство эксплуатации и обслуживания;
- 4.1) Уровень автоматизации;
- 4.2) Временная эффективность;
- 4.3) Ресурсоемкость;
- 5.1) Гибкость;
- 5.2) Мобильность;
- 5.3) Модифицируемость;
- 6.1) Полнота реализации;
- 6.2) Согласованность;
- 6.3) Логическая корректность;
- 6.4) Проверенность;
- 6.5) Защищенность.

В данном стандарте используются следующие понятия и определения:

**Программы** – это данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

**Программное средство** – объект, состоящий из программ, процедур, правил и документов, относящихся к функционированию системы обработки информации.

**Программный продукт** – это программное средство, предназначенное для поставки, передачи, продажи пользователю.

**Жизненный цикл (ЖЦ) программного средства или системы** – это совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение ПС или системы, охватывающая жизнь ПС или системы от установления требований к ним до прекращения их использования.

### 3.5. Рассмотрение метода количественной оценки качества программных средств

1. На фазе анализа проводится выбор показателей и их базовых значений.
2. Для показателей качества на всех уровнях принимается единая шкала оценки (от 0 до 1).
3. В процессе оценки качества на каждом уровне (кроме уровня оценочных элементов) проводится вычисление двух величин:

- абсолютного показателя качества  $P_{ij}$  ;
- относительного показателя качества  $R_{ij}$  ,

где  $j$  – порядковый номер показателя данного уровня для  $i$ -го показателя вышестоящего уровня.  $R_{ij} = \frac{P_{ij}}{P_{ij}^6}$

4. Каждый показатель качества второго и третьего уровней характеризуется двумя параметрами:

- количественным значением;
- весовым коэффициентом  $V_{ij}$ .

Сумма весовых коэффициентов всех показателей некоторого уровня, относящихся к показателю вышестоящего уровня, постоянна и равна 1.

5. Определение усредненной оценки  $m_{kq}$  оценочного элемента по нескольким его значениям (измерениям)  $m_{qt}$  осуществляется по следующей формуле  $m_{kq} = \frac{1}{T} \sum_{t=1}^T m_{qt}$

6. Итоговая оценка  $k$ -ой метрики  $j$ -го критерия определяется по формуле  $P_{jk}^M = \frac{1}{Q} \sum_{q=1}^Q m_{kq}$

7. Абсолютные показатели  $j$ -го критерия  $i$ -го фактора вычисляются по формуле  $P_{ij}^K = \sum_{k=1}^n (P_{jk}^M * V_{jk}^M) / \sum_{k=1}^n V_{jk}^M$

8. Относительные значения  $R_{ij}^K$  j-го критерия i-го фактора  $P_{ij}^K$  по отношению к базовому значению  $P_{ij}^{Ka}$  определяются по формуле:  $R_{ij}^K = P_{ij}^K / P_{ij}^{Ka}$

9. Абсолютные и относительные значения i-го фактора качества определяются по формуле  $P_i^\Phi = \sum_{j=1}^N (P_{ij}^K * V_{ij}^K) / \sum_{j=1}^N V_{ij}^K$

10. Общая оценка качества в целом формируется экспертами по набору полученных значений факторов качества.

**Характеристика качества программного средства** – категория свойств программного средства, с помощью которого выражается его качество. Характеристики качества ПС могут быть определены с помощью подхарактеристик и, в конечном итоге, атрибутов качества ПС.

**Подхарактеристика качества программного средства** - характеристика качества программного средства, входящая в состав другой характеристики качества.

Модель сопровождаемости для фазы анализа программных средств базируется на следующих процессах и фазах ЖЦ ПС:

### 3.6. Процесс разработки

1. Фаза анализа
2. Фаза проектирования
3. Фаза реализации
4. Фаза тестирования
5. Фаза изготовления

### 3.7. Процесс применения

1. Фаза внедрения
2. Фаза эксплуатации
3. Фаза сопровождения

Фазы представляют собой временные периоды, соответствующие работам, совокупностям работ или процессам жизненного цикла программных средств.

Выбор оценочных элементов метрики зависит от функционального назначения и формируется с учетом полученных ранее данных при проведении испытаний программного средства.

Для выбора оценочных элементов предполагается перечень таблиц содержащих наименование элементов, метод оценки. Чем больше уникальных модулей, тем и точнее показатель оценки.

### 3.8. Модель качества в использовании

**Качество в использовании (Quality in use)** – это способность программного продукта позволять заданным пользователям достигать заданные цели с результативностью, продуктивностью, безопасностью и удовлетворением в заданном контексте использования. Качество в использовании – это восприятие пользователем качества окружающей среды, содержащей ПП. Оно измеряется больше на основе результатов использования ПП в среде, чем на основе собственных внутренних свойств ПП.

Модель качества в использовании является **иерархической**. На ее верхнем уровне находятся четыре характеристики. Характеристики определяются непосредственно метриками (второй уровень модели). В отличие от модели внешнего и внутреннего качества уровень подхарактеристик в модели качества в использовании отсутствует.

Качество в использовании разделяется на четыре **характеристики**:

**1. Результативность** (Effectiveness) – это способность программного продукта, позволяющая пользователям достигать заданные цели с точностью и полнотой в заданном контексте использования.

**2. Продуктивность** (Productivity) – это способность программного продукта, позволяющая пользователям расходовать количество ресурсов, соответствующее результативности, достигаемой в заданном контексте использования. Ресурсы могут включать время выполнения задачи, усилия пользователя, материалы, стоимость использования.

**3. Безопасность** (Safety) – это способность программного продукта достигать приемлемых уровней риска причинения вреда людям, бизнесу, программному обеспечению, имуществу или окружающей среде в заданном контексте использования. Обычно риски – это результат дефектов в функциональности (включая защищенность), надежности, практичности и сопровождаемости.

**4. Удовлетворенность** (Satisfaction) – это способность программного продукта удовлетворять пользователя в заданном контексте использования. Удовлетворенность определяется реакцией пользователя на взаимодействие с программным продуктом и включает отношение к применению продукта.

Рассмотрев все описанное выше, мы можем перейти к разработке продукта оценки качества ПО.

## 4. Практическая часть

### 4.1. Анализ существующей методологии

Для создания своего приложения рассмотрим ГОСТ 28195-99.

На процессы разработки и оценки качества ПС оказывают влияние следующие обобщенные показатели ПС:

1. Область применения и назначение ПС;
2. Тип решаемых функциональных задач;
3. Объем и сложность ПС;
4. Необходимый состав и требуемые значения характеристик качества ПС и величина допустимого ущерба из-за недостаточного их качества;
5. Степень связи решаемых задач с реальным масштабом времени или допустимой длительностью ожидания результатов решения задачи;
6. Прогнозируемые значения длительности эксплуатации и перспектива создания множества версий ПС;
7. Предполагаемый тираж производства и применения ПС;
8. Степень необходимой документированности ПС.

В соответствии с ГОСТ 28195-99 под качеством любого изделия понимается совокупность свойств и характеристик изделия, относящихся к его способности удовлетворять установленные или предполагаемые потребности.

В литературе часто используется следующее определение качества ПС – это совокупность его свойств, обеспечивающая возможность достижения его целей без излишних затрат средств и труда со стороны пользователя.

Качество функционирования ПС – множество свойств, обуславливающих пригодность ПС обеспечивать надежное и своевременное представление требуемой информации потребителю для ее дальнейшего использования по назначению.

При оценке качества используются следующие типы шкал:

- 1) Номинальная – соответствует набору категорий;
- 2) Упорядоченная – соответствует упорядоченному набору делений шкалы;
- 3) Интервальная – соответствует упорядоченной шкале с равноудаленными делениями;
- 4) Относительная – соответствует упорядоченной шкале с равноудаленными делениями, оцененными в относительных единицах (относительно некоторой абсолютной величины).

Метрики, использующие номинальную и упорядоченную шкалы, применяются для оценки качественных показателей, которые нельзя измерить количественно. Метрики, использующие интервальную и относительную шкалы, применяются для оценки количественных показателей.

## 4.2. Определение спецификации

Анализируя ГОСТ 28195-99, определяем спецификации, т.е. формулируем, что нужно сделать:

- 1) определить значение оценочных характеристик и метрик;
- 2) определить значение критериев качества и факторов качества;
- 3) определить общую оценку качества;
- 4) разработать удобный интерфейс для работы с системой.

## 4.3. Определение структуры ПС

Проанализировав поставленную задачу, представим структурную схему создаваемого ПС на основе модульной декомпозиции (рис. 2).

Вся информация, собираемая для анализа ПС, будет храниться и представляться пользователю в виде таблицы, где будут описаны факторы, критерии качества, метрики и оценочные элементы. Для этого будет использован table widget. Пользователь может вносить показатели для каждого элемента расчета. Так же он может воспользоваться функцией рандомайзера для проверки работы ПС.

Рассмотрев все вышеизложенное можно приступить к разработке ПС.

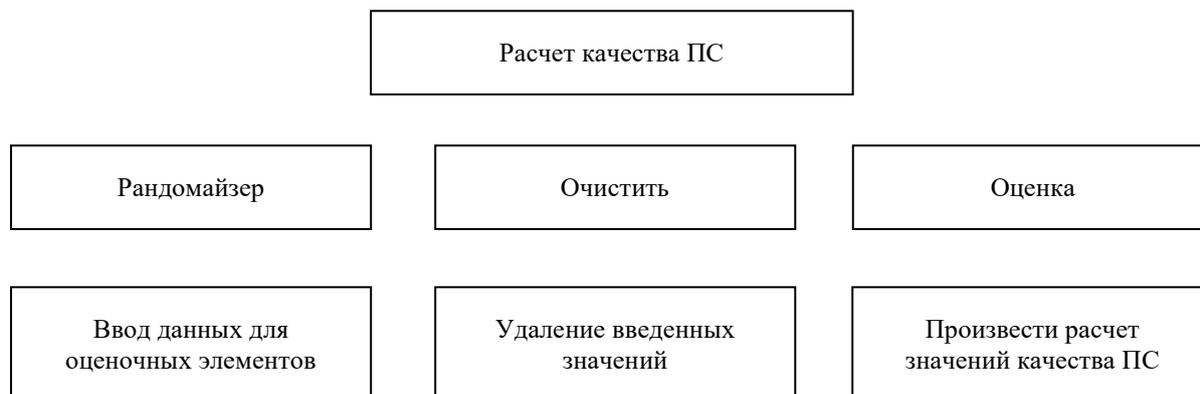


Рис. 2. Структурная схема программы для вычисления качества ПС в виде модульной декомпозиции

## 4.4. Описание работы ПС

После запуска ПС появляется основное диалоговое окно. Для работы с ПС доступны следующие функции:

- очистить значения (после нажатия кнопки «Очистить значения» выполняется процедура, которая удаляет значения);
- добавление случайных значений (нажмите кнопку «Рандомайзер» и виджете tableWidget у оценочных элементов появятся значения, сгенерированные функцией);
- оценка качества (после нажатия кнопки «Оценка» программа произведет расчет значений для каждого показателя).

## 4.5. Разработка пользовательского интерфейса

Первым шагом при разработке программы оценки ПС на Qt явилось создание интерфейса, с видимой частью приложения которого взаимодействует пользователь. Формы и элементы управления как раз и являются своего рода «строительными блоками» при создании интерфейса. Именно с этими объектами приходится работать при построении приложений. Формы – это объекты, которые обладают свойствами, определяющими их внешний вид, методами,

определяющими их поведение, и событиями, которые определяют их взаимодействие с пользователем. Мной был разработан интерфейс, представленный на рис. 3.

#### 4.6. Интерпретация полученных данных

Факторы	Оценка	Критерии качества	Оценка	Метрики	Оценка
Надежность		Устойчивость функционирования		Средство восстановления при при сбое оборудования	Возможность
Сопровождаемость		Работоспособность		Реализация управления средствами восстановления	Полнота обра
Удобство использования		Структурность		Функционирование в заданных режимах	Наличие сист данных
Эффективность		Простота конструкции		Обеспечение обработки заданного объема информации	Наличие сред входных данн
Универсальность		Наглядность		Использование основных логических структур	Наличие сред входных данн
Функциональность		Повторяемость		Соблюдение принципа нисходящего программирования	Наличие прои диапазону их
		Полнота документации		Простота архитектуры проекта	Наличие обр:
		Легкость освоения		Межмодульные связи	Наличие обр:
		Доступность программных документов		Комментарии логики программ проекта	Наличие мод
		Удобство эксплуатации и обслуживания		Оформление текста программ	Оценка прог: модулей
		Уровень автоматизации		Использование типовых компонентов ПС	Соблюдение сверху вниз
		Временная эффективность		Полнота документации фазы проектирования	Оценка прог:
		Ресурсоемкость		Полнота документации фазы реализации	Наличие ком машинозавис Наличие ком

Общая оценка

Рис. 3. Вид пользовательского интерфейса

После выполнения программы мы получаем результирующие данные для общей оценки качества ПС, а также результаты оценки метрик, критериев и факторов качества.

Показатели качества на каждом вышестоящем уровне (кроме уровня оценочных элементов) определяются показателями качества нижестоящего уровня, т. е.:

- результаты оценки каждого фактора определяются результатами оценки соответствующих ему критериев качества;
- результаты оценки каждого критерия определяются результатами оценки соответствующих ему метрик;
- результаты оценки каждой метрики определяются результатами оценки определяющих ее оценочных элементов.

Если результаты оценки попадают в диапазон от 1 до 0,9 – данному критерию присуждается оценка «отлично».

Если результаты оценки попадают в диапазон от 0,8 до 0,75 – данному критерию присуждается оценка «хорошо».

Если результаты оценки попадают в диапазон от 0,75 до 0,4 – данному критерию присуждается оценка «удовлетворительно».

Если результаты оценки попадают в диапазон от 0,4 до 0 – данному критерию присуждается оценка «плохо».

#### 4.7. Возможности улучшения и развития ПС

Данное программное средство, как и все, существующие на рынке, возможно улучшать, добавляя функционал и модернизировать под нужды заказчика.

В дальнейшем есть планы добавления функционала сохранения в файл. Это позволит делать более удобное представление результатов работы ПС, чтобы их можно было использовать в дальнейшем. Также есть планы добавления функционала считывания данных из файла, для возможности оптимизации и автоматизации процесса работы ПС.

В разработанном мной ПС прописаны не все критерии качества, метрики и оценочные элементы, описанные в ГОСТ 28195-99. Это дает направление расширения данного ПС. Так же некоторым заказчикам требуются другие показатели для оценки их продукта. ГОСТ 28195-99 позволяет модернизировать показатели, по которым происходит оценка ПС. И решение, разработанное мной, так же может быть модифицировано под решение конкретных задач. Мной предусмотрена возможность пересмотра весовых коэффициентов для расчета метрик под определенные запросы заказчиков.

### 5. Заключение

Повышения качества программных средств является основным фактором конкурентоспособности ПС на рынке. Объективная оценка качества ПС играет самую важную роль в этом процессе. Рынок программ, предоставляющих такой функционал мал, в связи с трудностью реализации проверки ПС, так как само ПО должно соответствовать всем стандартам.

Разработанная программа, на основе ГОСТа 28195-99, позволяет рассчитать качество ПС, но также может быть улучшена, путем добавления большего количества метрик и оценочных элементов для получения более точной оценки.

### Литература

1. Гиссин В.И. Управление качеством продукции: учеб. пособие/ В.И. Гиссин – Ростов на Дону: Феникс, 2007. 256 с.
2. ГОСТ Р ИСО 9001-2008. Системы менеджмента качества: утв. Приказом Ростехрегулирования от 18 декабря 2008 №471-ст/ 2008 «Ценообразование и сметное нормирование в строительстве» – №4 с. – М.: Стандартиформ, 2009. 46 с.
3. ГОСТ 28195–99 – 2000 Оценка качества программных средств. Общие положения. – Взамен ГОСТ 28195-89, введ. 2000-03-01 Мн.: Изд-во БелГИСС, 2004. 53 с.
4. Коротков Э.М. Концепция менеджмента. Москва, 2005. 18 с.
5. Б. Страуссгрупп. Язык программирования C++. Бином, 2011. 1136 с.

# Исследование возможностей API OpenAI

*А.Х. Исомадинов*

*Кузбасский государственный технический университет имени Т.Ф. Горбачева*

Данная статья описывает разработку AI-чат-бота на платформе Telegram для автоматизации процесса составления тестовых заданий на основе текста, отправленного пользователем. Бот создан на языке программирования Python, с использованием библиотек OpenAI и aiogram. В статье рассматриваются актуальность автоматизации процесса составления тестов, основные направления исследований в данном вопросе, а также описывается процесс разработки и функциональность AI-чат-бота. Результаты данного исследования могут быть полезны для улучшения организации рабочего процесса в образовательных учреждениях, а также для повышения качества контроля знаний в дистанционной форме обучения.

*Ключевые слова:* AI-чат-бот, автоматизация, составление тестов, контроль знаний, telegram, python, aiogram.

## 1. Введение

Важной составляющей процесса обучения является контроль знаний. В условиях, когда непрерывно возрастает доля дистанционных образовательных технологий, значительная часть контроля знаний осуществляется с применением тестирования. Составление тестов, используемых для проверки уровня освоения изученного материала, задача трудоемкая. Актуальность автоматизация процесса составления тестовых задания сложно переоценить. В результате анализа основных направлений исследований в данном вопросе [1] было принято решение о создании AI-чат-бота на платформе Telegram. Такие боты очень популярны среди пользователей и могут способствовать получению новых знаний и улучшению организации рабочего процесса. Рассматриваемый AI-чат-бот автоматизирует процесс составления тестовых вопросов на основе текста, отправленного пользователем. Для реализации используется язык программирования Python, библиотеки OpenAI и aiogram.

## 2. Создание чат-бота с библиотекой OpenAI

OpenAI [2] – платформа искусственного интеллекта, которая предоставляет API для создания мощных NLP-приложений. Одним из главных преимуществ OpenAI является доступность передовых технологий искусственного интеллекта и возможность использовать их для создания решений в различных сферах деятельности. Aiogram – это библиотека Python, которая упрощает создание чат-ботов Telegram.

Для работы необходимо получить токен для работы с Telegram и ключ API OpenAI. Последовательность создания бота представлена на рисунке 1.

Реализованы три обработчика:

1. Реагирует на команду «/start» и создает клавиатуру с тремя кнопками: «Тесты», «Вопросы» и «Смысл» (рисунок 2), а затем отправляет приветственное сообщение пользователю и переходит в состояние ожидания выбора.

2. Реагирует на нажатие одной из трех кнопок на клавиатуре и сохраняет выбранное пользователем действие в контексте. Затем бот отправляет запрос на ввод текста и переходит в состояние ожидания ввода текста пользователем (рисунок 3).

3. Обработчик ввода текстового сообщения пользователем в чат-бот, который использует API OpenAI для генерации ответа на основе введенного текста и выбранного действия.

Обработчик нажатия на кнопки использует лямбда-функцию, чтобы проверить, содержит ли сообщение текст, соответствующий одной из трех кнопок на клавиатуре, когда пользователь

находится в состоянии ожидания выбора. Это сделано для того, чтобы избежать конфликта с другими обработчиками, которые могут реагировать на сообщения в любое время.

Текст сохраняется в контексте с использованием библиотеки aiogram.

Затем формируется запрос к API OpenAI, в котором используется модель «text-davinci-003». Запрос зависит от выбранного действия. Если пользователь выбрал «Тесты», то в запросе указывается, что нужно составить 4 теста с вариантами А, В, С, D по тексту и в конце написать правильные ответы. Если пользователь выбрал «Вопросы», то в запросе указывается, что нужно составить 4 вопроса по тексту и ответы по вопросам. Если пользователь выбрал «Смысл», то в запросе указывается, что нужно сформировать краткое смысловое содержание.

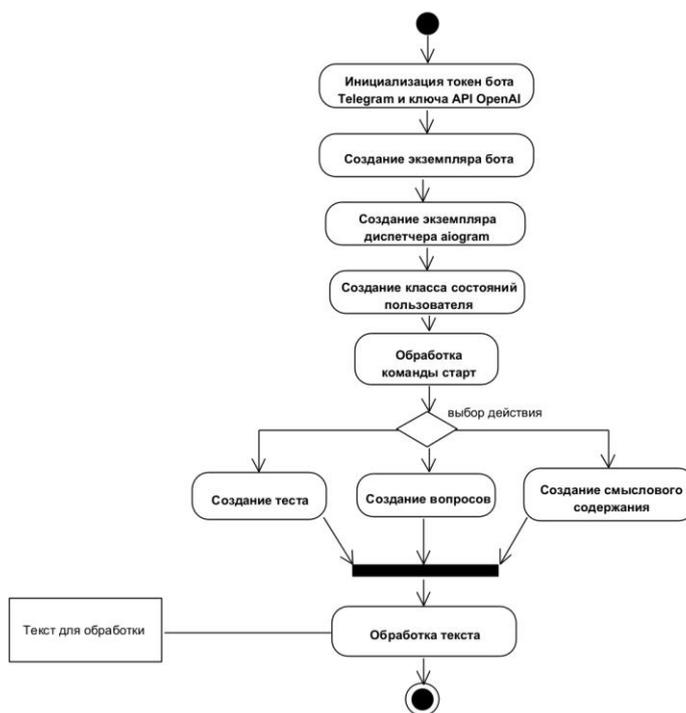


Рис. 1. Последовательность создания бота

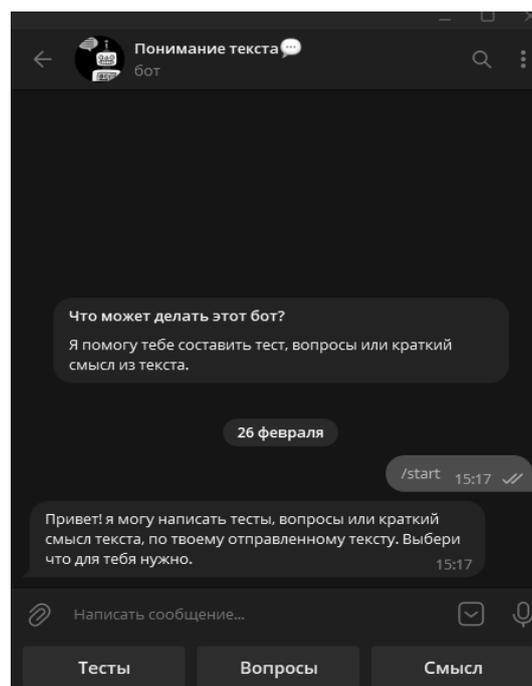
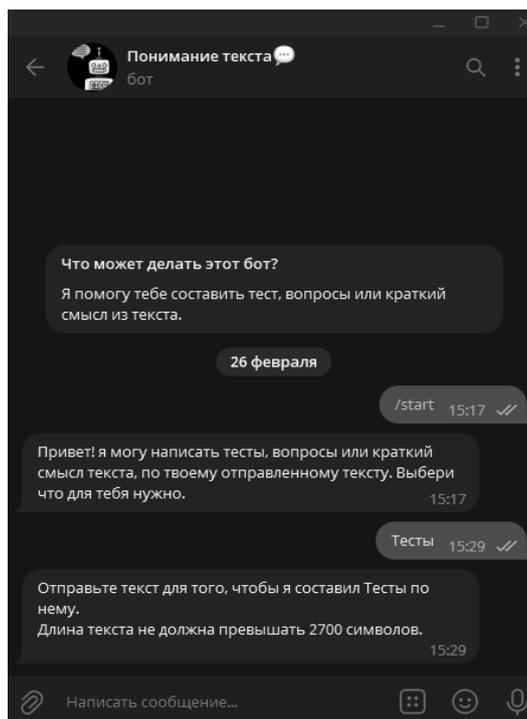


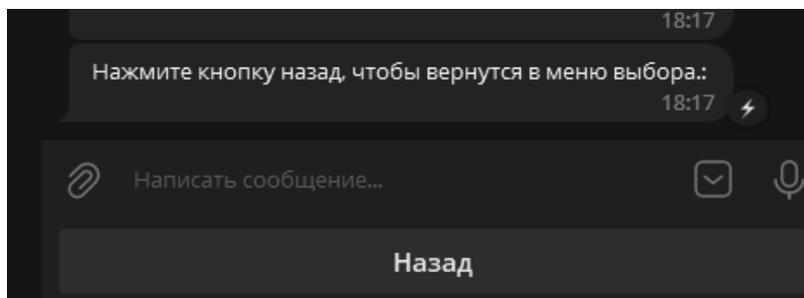
Рис. 2. Внешний вид бота после команды «/start»



**Рис. 3.** Внешний вид бота при запросе ввода текста

Для получения ответа от API OpenAI, используется библиотека `openai`. Ответ передается пользователю в виде сообщения.

По завершению формирования результата выполнения задания формируется кнопка «Назад» (на рисунке 1 не отображена), которая позволяет вернуться к выбору действий (рисунок 4).



**Рис. 4.** Возврат к выбору действий

Код, представленный на рисунке 5, запускает бесконечный цикл опроса Telegram API с помощью функции `start_polling` из библиотеки `aiogram`. Параметр `dp` указывает на экземпляр класса `Dispatcher`, который содержит обработчики для обработки входящих сообщений. Параметр `skip_updates=True` указывает на пропуск обновлений, которые пришли до запуска бота.

```
if __name__ == '__main__':
    executor.start_polling(dp, skip_updates=True)
```

**Рис. 5.** Код для опроса Telegram API

Конструкция `if __name__ == '__main__':` используется для того, чтобы код был выполнен только в случае, если модуль был запущен как главный файл программы, а не импортирован в другой модуль.

### 3. Пример работы чат-бота

Пример текста для обработки приведен на рисунке 6, результаты работы представлены на рисунках 7-9.

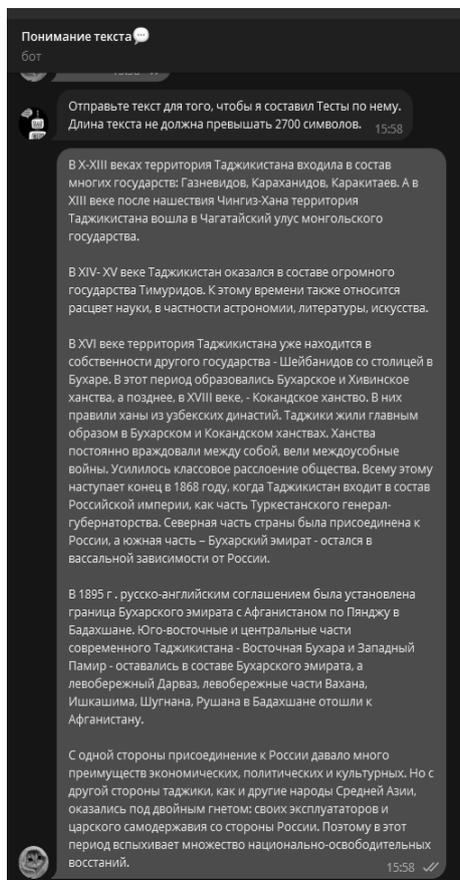


Рис. 6. Пример текста для обработки

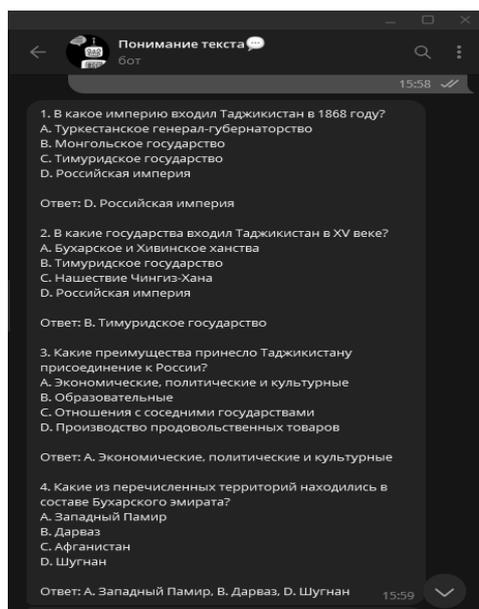


Рис. 7. Сформированные тестовые вопросы с вариантами ответов

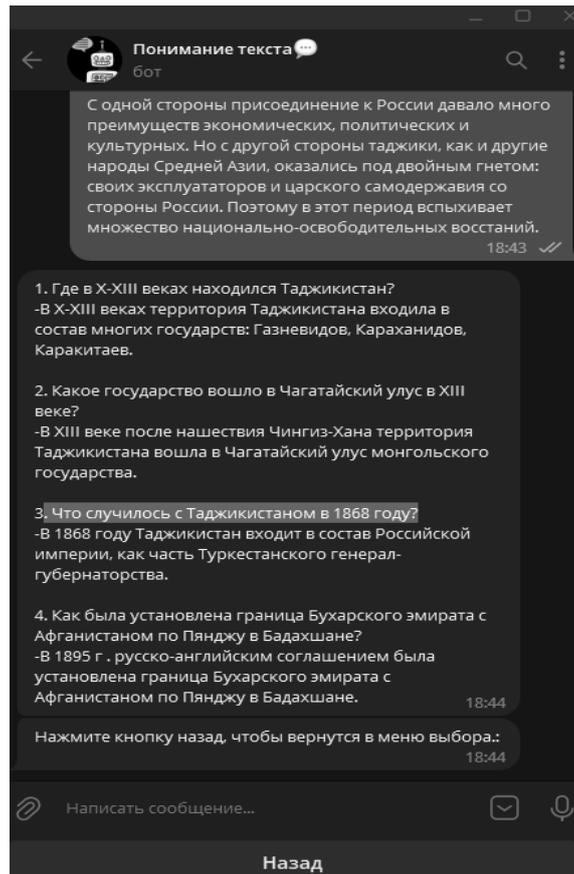


Рис. 8. Сформированные вопросы по тексту

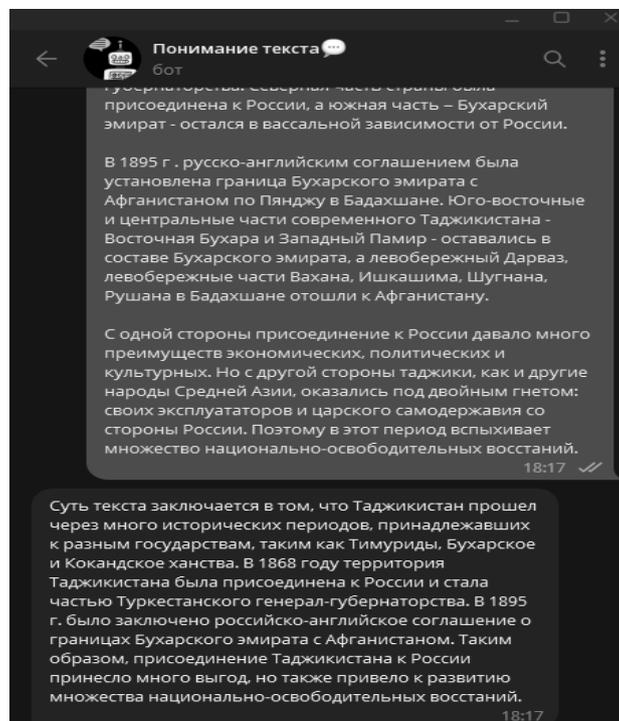


Рис. 9. Основная суть текста

## 4. Заключение

В результате работы над проектом были выявлены некоторые недостатки OpenAI:

1. Ограниченный доступ к функциям. Некоторые функции доступны только для платных пользователей, что может быть проблемой для небольших компаний и стартапов.

2. Ограниченные возможности интеграции. Некоторые инструменты OpenAI могут быть ограничены в интеграции с другими системами, что может вызвать проблемы при разработке решений, использующих несколько систем.

3. Безопасность и конфиденциальность. Использование искусственного интеллекта может привести к серьезным проблемам с конфиденциальностью и безопасностью данных, если не предпринимать соответствующие меры защиты.

4. Ограниченная точность. Как и любая система искусственного интеллекта, OpenAI не всегда дает точные результаты и может допускать ошибки. Это может стать проблемой при разработке высокоточных решений.

В данной статье было рассмотрено использование библиотеки OpenAI для генерации текстов на основе некоторых шаблонов и запросов от пользователя. Это открывает большие возможности для различных приложений, таких как генерация новостных статей, рекламных текстов и т.д. Также реализованное решение может быть использовано в учебном процессе для снижения нагрузки по составлению тестовых вопросов на преподавательский состав и в других ситуациях, когда необходимо оценить уровень полученных знаний.

Кроме того, рассмотрены преимущества и недостатки работы с OpenAI. Среди преимуществ можно выделить высокую точность генерации текстов, возможность работы с неструктурированными данными и простоту использования API. Однако, существуют и некоторые ограничения в применении: высокую стоимость при использовании всех функций, ограниченность доступа к API и необходимость в большом объеме обучающих данных для оптимальной работы моделей.

Таким образом, использование OpenAI для генерации текстов может быть очень полезным и эффективным, но также может требовать дополнительных ресурсов при выборе подходящей модели и настройке ее параметров.

## Литература

1. Швецов, А.Н., Куртасов, А.М. Метод автоматизированной генерации контрольно-тестовых заданий из текста учебных материалов [Электронный ресурс] // Вестник Череповецкого государственного университета. – 2014. – № 7 – Режим доступа: <https://cyberleninka.ru/article/n/metod-avtomatizirovannoy-generatsii-kontrolno-testovyh-zadaniy-iz-teksta-uchebnyh-materialov/viewer>, свободный. (Дата обращения 04.03.2023)
2. Официальная документация OpenAI [Электронный ресурс]. – Режим доступа: <https://beta.openai.com/docs/>, свободный. (Дата обращения 09.03.2023)
3. Hands-On Chatbots and Conversational UI Development, Srin Janarthanam [Электронный ресурс]. – Режим доступа: <https://www.packtpub.com/product/hands-on-chatbots-and-conversational-ui-development/9781801072978>, свободный. (Дата обращения 18.03.2023)
4. How to Build a Telegram Chatbot in Python with OpenAI's GPT-3, Twilio [Электронный ресурс]. – Режим доступа: <https://www.twilio.com/blog/how-to-build-a-telegram-chatbot-in-python-with-openais-gpt-3>, свободный. (Дата обращения 09.03.2023)

# Золотое сечение в фотографии

*И.А. Кириллова*

*Нижегородский филиал Самарского государственного университета путей сообщения*

В данной статье рассматривается возможность получения золотого сечения в фотографиях, основные правила золотого сечения, отношение людей к золотому сечению, построение пропорции.

*Ключевые слова:* движение, горизонт, правила ринга, встречные курсы, правило диагоналей, результаты опроса, второе золотое сечение.

## Введение

Геометрия владеет двумя сокровищами: одно из них – теорема Пифагора, другое – деление отрезка в среднем и крайнем отношении. И. Кеплер

Есть вещи, которые нельзя объяснить. Вот вы подходите к пустой скамейке и садитесь на нее. Где вы сядете — посередине? Или, может быть, с самого края? Нет, скорее всего, не то и не другое. Вы сядете так, что отношение одной части скамейки к другой, относительно вашего тела, будет равно примерно 1,62. Простая вещь, абсолютно инстинктивная... Садясь на скамейку, вы произвели «золотое сечение».

## 1. История создания «Золотого сечения»

Древнейшим литературным памятником, в котором встречается деление отрезка в отношении золотого сечения, являются "Начала" Евклида (III в. до н. э.). Золотое сечение было известно и до Евклида. В частности, знали о нем Пифагор, древнегреческий философ и математик (VI в. до н.э.). Есть предположение, что Пифагор свое знание золотого деления позаимствовал у египтян и вавилонян. И действительно, пропорции пирамиды Хеопса, храмов, барельефов, предметов быта и украшений из гробницы Тутанхамона свидетельствуют, что египетские мастера пользовались соотношениями золотого деления при их создании. Платон (427...347 гг. до н.э.) также знал о золотом делении. В дошедшей до нас античной литературе золотое деление впервые упоминается в "Началах" Евклида. Во 2-й книге "Начал" дается геометрическое построение золотого деления. После Евклида исследованием золотого деления занимались Гипсикл (II в. до н.э.), Папп (III в. н.э.) и др.

В эпоху Возрождения усиливается интерес к золотому делению среди ученых и художников в связи с его применением как в геометрии, так и в искусстве, особенно в архитектуре. В это время появилась книга монаха Луки Пачоли. В 1509 г. в Венеции была издана книга Луки Пачоли "Божественная пропорция" с блестяще выполненными иллюстрациями, ввиду чего полагают, что их сделал Леонардо да Винчи. Книга была восторженным гимном золотой пропорции. Леонардо да Винчи также много внимания уделял изучению золотого деления. Он производил сечения стереометрического тела, образованного правильными пятиугольниками, и каждый раз получал прямоугольники с отношениями сторон в золотом делении. Поэтому он дал этому делению название золотое сечение. В то же время в Германии, над теми же проблемами трудился Альбрехт Дюрер. Альбрехт Дюрер подробно разрабатывает теорию пропорций человеческого тела. Важное место в своей системе соотношений Дюрер отводил золотому сечению. Известен пропорциональный циркуль Дюрера. Великий астроном XVI в. Иоганн Кеплер назвал золотое сечение одним из сокровищ геометрии. Он первый обращает внимание на значение золотой пропорции для ботаники (рост растений и их строение).

В последующие века правило золотой пропорции превратилось в академический канон. Вследствие вновь "открыто" золотое сечение было в середине XIX в. В 1855 г. немецкий исследователь золотого сечения профессор Цейзинг опубликовал свой труд "Эстетические исследования". Он абсолютизировал пропорцию золотого сечения, объявив ее универсальной для всех явлений природы и искусства. Цейзинг дал определение золотому сечению, показал, как

оно выражается в отрезках прямой и в цифрах. Когда цифры, выражающие длины отрезков, были получены, Цейзинг увидел, что они составляют ряд Фибоначчи, который можно продолжать до бесконечности в одну и в другую сторону.

На протяжении многих веков, для построения гармоничных композиций художники пользуются понятием "Золотого сечения". Обнаружено, что определенные точки в картинной композиции автоматически привлекают внимание зрителя. Таких точек всего четыре соответствующих краев плоскости. Разумеется, в момент съемки мы не в состоянии просчитать и зрительно отложить в уме необходимые пропорции. Поэтому на момент съемки используется упрощенный вариант построения «Золотого сечения» или правило «Трети». Заключается оно в следующем: мы мысленно делим кадр на три части по горизонтали и вертикали и, в точках пересечения воображаемых линий, размещаем ключевые детали снимаемой сцены.

Меня заинтересовало, как сделать фотографию интересной, выразительной, притягивающей взгляды зрителей. Для создания фотографии недостаточно только снять изображение. Необходимо гармонично разместить объекты на снимке, наполнив его смыслом. Существуют разные способы и правила для создания гармоничной композиции. Иногда достаточно разместить объекты съемки в определенных местах. В других случаях для этого достаточно правильно выбрать точку съемки. Небольшое смещение положения фотоаппарата может внести существенные изменения в композицию. Для придания выразительности вашим фотографиям, применяйте правила построения композиции.

## 2. Второе золотое сечение

Болгарский журнал «Отечество» (№10, 1983 г.) опубликовал статью Цветана Цекова-Карандаша «О втором золотом сечении», которое вытекает из основного сечения и дает другое отношение  $44 : 56$ . Такая пропорция обнаружена в 9 архитектуре, а также имеет место при построении композиций изображений удлиненного горизонтального формата.

Деление осуществляется следующим образом. Отрезок АВ делится в пропорции золотого сечения. Из точки С воссоставляется перпендикуляр CD. Радиусом АВ находится точка D, которая соединяется линией с точкой А. Прямой угол ACD делится пополам. Из точки С проводится линия до пересечения с линией AD. Точка E делит отрезок AD в отношении  $56 : 44$ .

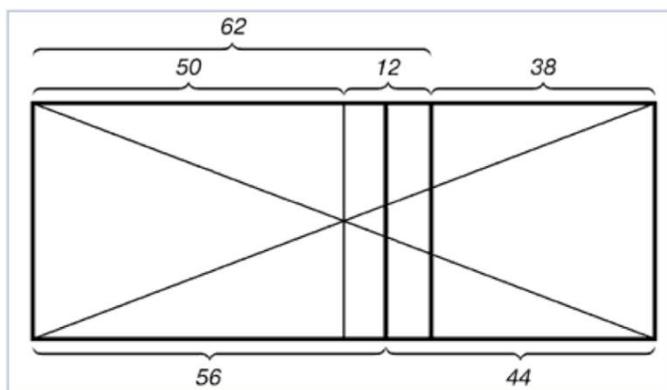


Рис. 1. Деление прямоугольника линией второго золотого сечения

На рисунке показано положение линии второго золотого сечения. Она находится посередине между линией золотого сечения и средней линией.

## 3. Построение пропорции

Практическое знакомство с золотым сечением мы начали с деления отрезка прямой в золотой пропорции с помощью циркуля и линейки. Из точки В воссоставляется перпендикуляр, равный половине АВ. Полученная точка С соединяется линией с точкой А. На полученной линии откладывается отрезок ВС, заканчивающийся точкой D. Отрезок AD переносится на прямую АВ.

Полученная при этом точка E делит отрезок АВ в соотношении золотой пропорции. Отрезки золотой пропорции выражаются бесконечной иррациональной дробью  $AE = 0,618\dots$ , если АВ принять за единицу,  $BE = 0,382\dots$ . Используют приближенные значения 0,62 и 0,38.

Свойства золотого сечения описываются уравнением:  $x^2 - x - 1 = 0$ . Решение этого уравнения:  $x_{1,2} = \frac{1 \pm \sqrt{5}}{2}$ .

## Деление отрезка прямой по золотому сечению

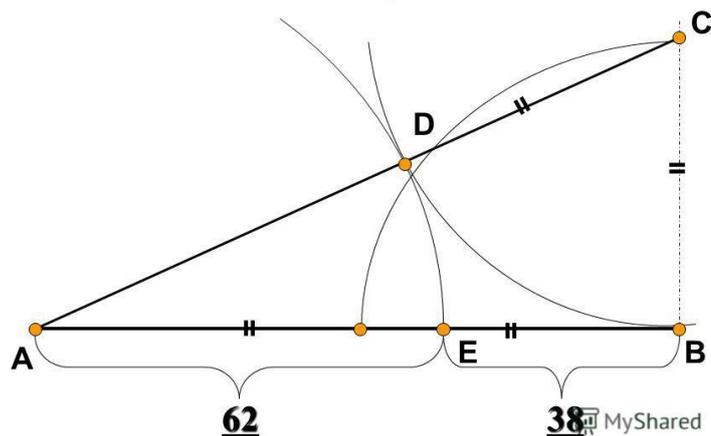


Рис. 2. Деление отрезка прямой по золотому сечению.  $BC = 1/2 AB$ ;  $CD = BC$  прямоугольника

**Движение.** Используется, когда вы хотите передать движение предмета. Оставляйте в кадре перед движущимся объектом больше места, чем у него «за спиной», т.е. располагаем объект на правой вертикальной линии золотого сечения. То же самое касается взгляда. Оставляйте перед взглядом больше места.

**Горизонт.** Он должен проходить в районе верхней или нижней линии золотого сечения. Когда линия горизонта делит кадр пополам - это чаще всего плохо смотрится.

«Правило ринга». Бывает, что объектов съёмки два. В большинстве случаев их лучше всего развести по противоположным «узлам золотого сечения», как боксёров по рингу. Поэтому не стесняйтесь, если чувствуете, что объект съёмки лучше немного сместить, - сместите.

«Встречные курсы». На фотографии объекты двигаются навстречу друг другу. Наше сознание помогает им. Поэтому в кадре, где объекты сближаются лучше расположить их немного дальше друг от друга. Сознание смотрящего на снимок сблизит их само. В случае, если объекты «расходятся», ситуация обратная.

**Правило диагоналей.** Траектория движения фокуса внимания зрителя редко бывает параллельно краям кадра, поэтому расположение отдельных линий сцены по диагоналям кадра воспринимается гармонично и позволяет связать содержимое кадра (например, передний и задний план) в единое целое.

- Держите камеру на уровне объекта съёмки. Не фотографируйте прямо снизу вверх или с высоты вашего роста вниз, кроме случаев, когда вы хотите добиться особого эффекта. Например, если Вы снимаете детей, опуститесь до уровня их глаз, иначе у вас получатся искаженные пропорции.

- Если вы снимаете движущийся объект, то оставляйте на фотографии пространство перед объектом, то есть по ходу его движения. Другими словами, располагайте объект, как будто он только зашёл на фотографию, а не покидает её.

- Старайтесь добиться того, чтобы источник света был сзади вас. А также избегайте ярких огней или пестрых цветных пятен в стороне от главного сюжета. Это отвлекает зрителя.

- Следите, чтобы главный объект снимка не сливался с фоном. Если вы снимаете какой-то один объект, то старайтесь выбирать простой фон, детали которого не будут отвлекать зрителя.

В некоторых случаях имеет смысл сделать так, чтобы объект занимал подавляющее большинство площади самого кадра, но утяжелять снимок нельзя.

- Попробуйте сделать сбалансированную композицию, так чтобы верхняя часть фотографии не выглядела "тяжелее", чем нижняя. Данное правило относится и к сторонам изображения.

Результаты опроса: 78% опрошенных проголосовали за фотографии с золотым сечением, а 22% за фотографии без золотого сечения. Таким образом, мы видим, что люди интуитивно выбирают фотоснимки, сделанные по принципу Золотого сечения. Это еще раз подтверждает, что наличие золотого сечения придает фотографии некую законченность и гармоничность.

## **Литература**

1. Воробьев Н.Н. Числа Фибоначчи. М.: Наука, 1964. 144 с.
2. Д. Пидоу. Геометрия и искусство. М.: Мир, 1979. 332 с.
3. Журнал «Квант». 1973, № 8.
4. Журнал «Математика в школе», 1994.
5. Ковалев Ф.В. Золотое сечение в живописи. К.: Высшая школа, 1989. 143 с.
6. Стахов А.П. Коды золотой пропорции. М.: Радио и связь, 1984. 152 с.

# Прикладное применение математики в программировании

*Т.В. Кузнецова*

*Поволжский государственный технологический университет*

Что же такое прикладная математика? Прикладная математика – это применение математических методов и алгоритмов в самых различных сферах жизнедеятельности человека. Применение математики к любой практической проблеме начинается с построения математической модели. Казалось бы, где здесь связь между математикой и программированием? В работе раскрывается применение математических методов в самых азах написания программы.

*Ключевые слова:* математика, программирование, JAVA, прикладное применение, булева алгебра.

## 1. Введение

Сокращение от «Information Technology» – IT, что с английского переводится как Информационные Технологии, быстро вошло в нашу жизнь, речь, мысли. Информационные технологии – это процессы, которые используют совокупность средств и методов сбора, обработки и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса или явления (информационного продукта). В наше время происходит бурное развитие, распространение цифрового пространства, знаний в этой области.

Казалось бы, велика ли здесь роль такой точной науки как математики? Важна ли программисту математика? На этот вопрос можно ответить совершенно по-разному. Математика помогает нам не только в решении арифметических задач. Благодаря ей развивается гибкость ума, что в последствии помогает объективно решить задачи любой сложности. Так, значение математики описывал М.В. Ломоносов: «Математику изучать надобно, поскольку она в порядок ум приводит».

Царица всех наук применяется во многих областях программирования. Не так сложно попасть в IT-сферу без глубоких знаний математики, но это не означает, что человек сможет стать профессионалом области.

Для того, чтобы программа была действительно полезной и хорошо работала, необходимо продумать все до мельчайших подробностей. В процессе её написания необходимо будет исследовать многие закономерности, следствия, делать какие-либо выводы и принимать решения на их основе. При написании программы необходимо понимать, что даст тот или иной метод, как поведёт себя программа при изменении какого-либо значения или параметра. Для этого необходимо логика. Это самая главная и отличительная черта программистов.

**Цель:** Выяснить прикладное применение математики в основах программирования.

**Задачи:**

1. Рассмотреть взаимосвязь точной науки и кода.
2. Ответить на вопрос: “Какие умения нужны программисту?”
3. Выяснить какие разделы математики применяются в написании кода.
4. Рассмотреть на примерах.

**Методы:**

1. Изучение литературы и других источников информации (в том числе СМИ и информации в сети Internet).
2. Анализ статистики.
3. Сравнение.

## 2. Основная часть

### 2.1 Теоретическая справка

Отвлечение от основной задачи и сведение её к более простым – главное, что связывает математику и программирование. Благодаря такому подходу становится возможным разложить задачу в форме кода.

Основываясь на этом, можно сделать вывод о том, какими умениями должен обладать программист:

- пониманием контекста задачи, возможностью убрать всё лишнее, оставляя самое главное;
- навыками разделения сложной системы на подсистемы, отражающие процесс.

Математическая наука и написание кода имеют общее основание – логику. Без специализированных знаний не так сложно стать тестировщиком, штатным программистом. Математика упрощает исчисление и построение программ. Хороший IT-специалист вынужден и обязан сам вводить, выводить алгоритмы для каждой отдельной задачи.

Для разных видов программирования нужно понимание разных разделов математики. Например, дифференциальные уравнения и геометрия пригодятся вам в графике. Без дискретной математики вы не сможете написать базы данных, создать поисковые системы или даже построить маршрут. Для разработки компьютерных игр, вам придется изучить все разделы математики, ведь там нужно заниматься моделированием физических процессов, создавать искусственный интеллект, обрисовывать графику. Линейная алгебра помогает разработчикам игр в перемещении в игровом пространстве персонажей, смене положения камеры, кнопок и мыши.

Наконец, последний пример: использование линейной алгебры при построении игр. Зачем нам линейная алгебра? Одним из направлений в линейной алгебре является изучение векторов. Если в вашей игре применяется работа с камерой и её направлением, скоростями объектов, то вам придётся иметь дело с векторами. В играх вектора используются для хранения местоположений, направлений и скоростей.

При проектировании приложения важную роль занимает математика. Без неё просто невозможно будет рассчитать поведение объектов, невозможно использовать какие-либо функции и методы. Если вы собираетесь стать программистом, то вам нужно понимать хотя бы основы.

### 2.2 Логика

Булева алгебра (или алгебра логики) используется почти во всех языках программирования. Это элементарные основы, без которых невозможно изучить принцип написания кода и выражений. Именно поэтому я считаю нужным изучить эту область более подробно.

Этот раздел назван в честь Джорджа Буля, который показал, как из любого числа высказываний, включающих любое число терминов, вывести любое заключение, следующее из этих высказываний, путем чисто символических манипуляций. Булева логика помогает в изучении методов определения истинности и ложности на языке компьютера и проста для самостоятельного изучения.

Булевым выражением называется такое выражение, которое принимает два значения: False (0 – ложь) или True (1 – истина). Для того, чтобы в зависимости от ситуации выполнялись разные инструкции (то есть чтобы программа не была линейной), используют булевы выражения. Результат же используется для определения пути кода.

Сложные логические выражения формируются благодаря логическим операторам. Логическими операторами являются: конъюнкция (логическое умножение) – *and*, дизъюнкция (логическое сложение) – *or*, отрицание (инверсия) – *not*.

Условные операторы используются для выполнения различных действий в зависимости от того, выполняется ли какое-либо условие. Условие – выражение, принимающее значение *true* или *false*.

Также стоит упомянуть, что наши компьютеры запрограммированы двоичным кодом, то есть любой символ представлен в виде нулей и единиц.

## 3. Практическое применение

### 3.1 Алгебра логики

Рассмотрю на примере использование алгебры логики, используя язык программирования Java.

Оператор *if* является одним из наиболее часто используемых условных операторов. Если выражение условия оператора *if* истинно (имеет значение *true*), то выполняется блок кода внутри оператора *if*. Если выражение оказывается ложным, выполняется первый набор кода после окончания оператора *if* (после закрывающей фигурной скобки).

#### Синтаксис:

```
if (условие) {  
    //блок команд  
}
```

**Задача 1.** Даны два числа 8 и 5. В случае, если первое число больше второго, выводится их разница, в противном случае их сумма.

```
public class Main  
{  
    public static void main(String[] args) {  
        int a=8, b=5;  
        if (a > b ) {  
            System.out.print(a-b);  
        }  
        else {  
            System.out.print(a+b);  
        }  
    }  
}
```

Рис. 1. Пример кода для решения задачи 1

*Пояснение:* На входе даны два целых действительных числа. Далее оператор сравнения проверяет больше ли первое число второго. Если утверждение истинно, выводит их разницу. Если утверждение ложно, то считает их сумму.

**Задача 2.** На вход подаётся целое число. Выведите "YES", если оно нацело делится на 7, но не делится на 5, иначе – "NO".

```
import java.util.Scanner;  
class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        int a = sc.nextInt();  
        if (((a%7)==0) & ((a%5)!=0)) {  
            System.out.print("YES");  
        } else {  
            System.out.print ("NO");  
        }  
    }  
}
```

Рис. 2. Пример кода для решения задачи 2

По аналогии с первой задачей стоит условие, при выполнении которого выводятся YES или NO соответственно. Так, если число, введённое пользователем, кратно 7, и при делении этого числа на 5 остаток не равен нулю, выводится "YES" («ДА»). В противном случае – "NO" («НЕТ»).

В каждом языке программирования, даже самом простом, есть оператор ветвления, позволяющий проверить истинность логического выражения и, в зависимости от его результата, выполнить то или иное действие.

Таким образом, в ходе работы программы используется алгебра логики, а именно булевские выражения, благодаря которым компьютер понимает, что должно происходить.

## 4. Заключение

Все достаточно сложные математические концепции основываются на логике и символическом «языке», который функционирует по определённому набору правил. А это очень похоже на само программирование. Чем больше продвинутых разделов математики знает человек, тем проще ему заниматься программированием. Но не потому, что где-то в программировании напрямую нужно применять, а потому что и там, и там — крайне похожие процессы. Опыт из одного легко перенести в другой, и наоборот.

Знания в области математики позволяют писать программисту более эффективные коды, выстраивать заранее эффективность еще не написанных алгоритмов и лучше описывать объекты реального мира. Именно это показано в данной работе на уровне элементарных понятий.

## Литература

1. Акулич И.Л. Математическое программирование в примерах и задачах. Минск: Высшая школа, 2009.
2. Владимиров Д.А. Булевы алгебры. М: «Наука», 1969. 320 с.
3. Копченова Н.В., Марон И.А. Вычислительная математика в примерах и задачах: Учебное пособие. 2-е изд., стер. СПб.: «Лань», 2008. 368 с.
4. <https://fb.ru/article/426683/matematika-dlya-programmista---osobnosti-razdelyi-i-rekomendatsii> [Электронный ресурс].
5. [itmathrepetitor.ru/zachem-programmistu-vysshaya-matem/](http://itmathrepetitor.ru/zachem-programmistu-vysshaya-matem/) [Электронный ресурс].

# **Применение имитационной модели искусственного интеллекта для автоматизированной оптимизации процесса разработки угольных месторождений**

*Р.В. Майтак, П.А. Пылов, А.В. Дягилева*

*Кузбасский государственный технический университет имени Т. Ф. Горбачева*

Разработка угольных месторождений – идеальная среда для применения интеллектуальных систем, основанных на моделях искусственного интеллекта, поскольку нахождение пластов полезных ископаемых порой не представляется возможным алгоритмам прямого программирования. В статье будет представлен пример автоматизированной разработки угольного месторождения с графической средой воспроизведения результатов – данная визуализация позволяет достичь правильного соотношения графической и аналитической информации, а также реализовать гибкую интеграцию модели с данными конкретного промышленного предприятия.

*Ключевые слова:* шахта, разработка угольных месторождений, искусственный интеллект, прикладное глубокое обучение.

## **1. Введение**

Добыча полезных ископаемых шахтным образом – трудоемкий процесс. Когда подземный промышленный комплекс функционирует и налажен производственный процесс добычи, действия в нем можно сравнить с итерациями цикла в языке программирования: они повторяются по заранее определенной схеме.

Однако, эта схема, в случае с добывающими шахтными комплексами, требует сложной проработки. Ее составление является результатом исследования месторождений полезных ископаемых в окончательном, сформированном специалистами, плане.

Появление утвержденного плана обязано множеству затраченных ресурсов – это не только аппаратная, но и административная составляющая.

Организация автоматизированной разработки угольных месторождений шахтным способом – очень ресурсоемкая задача. На уровень качества ее реализации влияет очень много параметров: начиная от выбора исходного математического аппарата, на основе которого будут вычисляться базовые формулы, и заканчивая человеческими ресурсами. Особенность человеческого труда состоит и в том, что люди могут допустить ошибку, которую может быть очень сложно оперативно исправить.

Существование такой интеллектуальной системы, которая бы позволяла объединить в себе комплексы математического, аналитического, графического и других аппаратов в единой программной оболочке позволило бы значительно упростить задачи проектирования шахтных комплексов. Кроме этого, внедрение интеллектуальных систем в такие информационные среды позволяет решать не только задачу автоматизации, но и одновременно с ней, оптимизировать разрабатываемую систему горных выработок.

## **2. Интеллектуальные системы в задаче имитационного моделирования добычи полезных ископаемых**

Превосходство и эффективность таких систем, в сравнении с обычными программируемыми, подтверждена в других прикладных сферах: например, в турнирах по шахматам, го, покеру, где ключевую роль оптимизации играет не только умение применять математические законы на практике, но и глубина расчетов [1]. В поединках между стандартными информационными программами и интеллектуальными системами, лидерство стабильно одерживают вторые [2].

Цель представленной в рамках данной статьи сети искусственного интеллекта – имитировать действия подразделений шахтной бригады с учетом запасов полезных ископаемых, находящихся под поверхностью участка разработки (обозначим такую поверхность «рабочей доской» или сокращенно «доской»).

Архитектура разрабатываемой интеллектуальной модели схожа с той, которую использовала команда HandyRL в Hungry Goose, решавшая задачу другой прикладной области знаний [3].

Количество необходимых, первоначально задаваемых параметров, определяются пользователем. Например, это могут быть:

- Тип самостоятельной единицы специальной техники;
- Количество самостоятельных единиц специальной техники каждого типа;
- Грузоподъемность каждой единицы;
- Число сотрудников в подразделении;
- Уровень запасов полезного ископаемого (предполагаемый).

```
def input_processing(obs, unit_id):
    width, height = obs['width'], obs['height']
    x_shift = (32 - width) // 2
    y_shift = (32 - height) // 2
    cities = {}
    b = np.zeros((20, 32, 32), dtype=np.float32)
    for update in obs['updates']:
        strs = update.split(' ')
        input_identifier = strs[0]
        if input_identifier == 'u':
            x = int(strs[4]) + x_shift
            y = int(strs[5]) + y_shift
            wood = int(strs[7])
            coal = int(strs[8])
            uranium = int(strs[9])
            if unit_id == strs[3]:
                # Текущая позиция и размер полезного груза залежей
                b[:2, x, y] = (1, (wood + coal + uranium) / 100)
            else:
                # Единицы измерения
                team = int(strs[2])
                cooldown = float(strs[6])
                idx = 2 + (team - obs['player']) % 2 * 3
                b[idx:idx + 3, x, y] = (1, cooldown / 6, (wood + coal + uranium) / 100)
        elif input_identifier == 'ct':
            team = int(strs[1])
            city_id = strs[2]
            x = int(strs[3]) + x_shift
            y = int(strs[4]) + y_shift
            idx = 8 + (team - obs['player']) % 2 * 2
            b[idx:idx + 2, x, y] = (1, cities[city_id])
        elif input_identifier == 'r':
            # Ресурсы
            r_type = strs[1]
            x = int(strs[2]) + x_shift
            y = int(strs[3]) + y_shift
            amt = int(float(strs[4]))
            b[{'wood': 12, 'coal': 13, 'uranium': 14}[r_type], x, y] = amt / 800
        elif input_identifier == 'rp':
            # Исследовательские точки входа (при наличии определенных наработок
            # для улучшения работы системы)
            team = int(strs[1])
            rp = int(strs[2])
            b[15 + (team - obs['player']) % 2, :] = min(rp, 200) / 200
        elif input_identifier == 'c':
            city_id = strs[2]
            fuel = float(strs[3])
            lightupkeep = float(strs[4])
            cities[city_id] = min(fuel / lightupkeep, 10) / 10

    # Дневная/Ночная смена для подсчета эконометрики
    b[17, :] = obs['step'] % 40 / 40
    # Повороты при разработке (настраиваемый опциональный критерий)
    b[18, :] = obs['step'] / 360
    # Размер карты
    b[19, x_shift:32 - x_shift, y_shift:32 - y_shift] = 1
    return b
```

Рис. 1. Листинг главной функции организации свертки в нейронной сети имитационной модели

После этого, интеллектуальная система первоначально преобразует исходные параметры (отформатировав их) в 32 канала и слой свертки. Далее это переводится в 12 слоев внутри сверточной нейронной сети с фильтрами 3x3, пропускными соединениями и пакетной нормализацией [1–3].

Во всех слоях размеры рабочей доски остаются постоянными, и прогнозирование выполняется для объектов в координате единицы измерения/города с полностью подключенным слоем. Предусмотрено пять возможных действий, по одному для каждого из четырех направлений и одно для собственного проектирования шахтного комплекса.

Интеграция основы представленной программной схемы организации модели искусственного интеллекта представлена в виде листинга программного кода на высокоуровневом языке программирования Python [4] на рисунке 1.

Основанная модель позволяет загрузить среду очень качественной графической прорисовки (рисунок 2 – 5), в которой уже присутствует видеоряд для удобного анализа временных, финансовых и других затрат на создание комплекса. Кроме того, создание комплекса максимально оптимизировано под местонахождение пластов полезных ископаемых, что позволяет в значительной степени снижать уровень множества затрат одновременно.



Рис. 2. Определение месторождений полезных ископаемых и их нанесение на слой карты

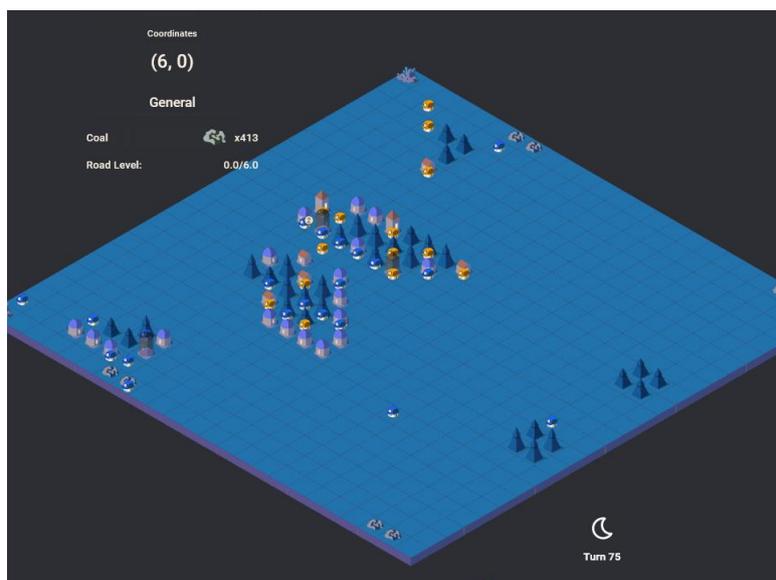
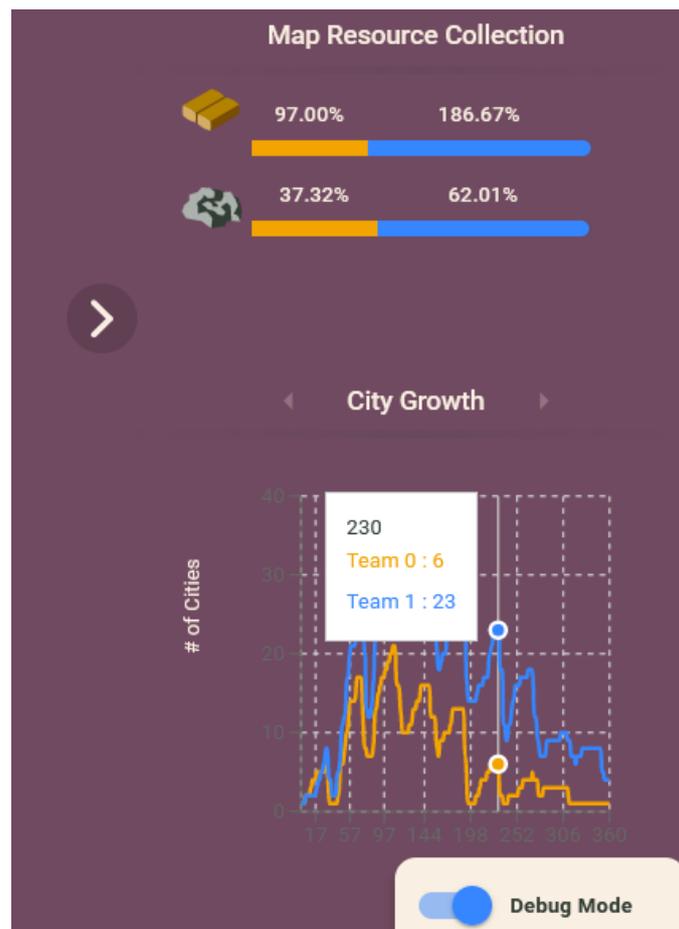


Рис. 3. Представление ночной смены графическим интерфейсом среды имитационной модели



**Рис. 4.** Уменьшение запасов полезного ископаемого (параметр coal пребывает в коэффициенте x83 по сравнению с x418 и x413 рисунков 2 и 3 соответственно)



**Рис. 5.** Итоговая статистика производительности команд и суммарный объем добычи полезных ископаемых (с соотношением долей на каждую бригаду рабочих)

Графическая визуализация встроенной в имитационную модель параметров позволяет не только адаптивно подстраивать нужные режимы (дневная/ночная смена, отрисовка запасов угля на каждом слое и так далее), но и позволяет использовать результаты функционирования интеллектуальной системы в качестве презентационного материала деятельности шахтного комплекса.

## Литература

1. Michael Nielsen. Neural Networks and Deep Learning. Apress, 2019. 224 с.
2. Проект команды HandyRL [Электронный ресурс]. – Режим доступа: <https://vc.ru/flood/39184-istoriya-sorevnovaniy-ii-i-cheloveka-kto-kogo> .
3. Проект соревнования команды HandyRL [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/code/yuricat/smart-geese-trained-by-reinforcement-learning/notebook> .
4. Пылов П.А., Майтак Р.В., Дягилева А.В. Разработка интеллектуальных систем для обработки сигналов с датчиков давления. Инфра-Инженерия, 2023. 155 с.

# Разработка мобильного приложения для диагностики и лечения компьютерной зависимости

*Д.М. Моисеев*

*Нижегородский филиал Самарского государственного университета путей сообщения*

Огромное количество информации и времени проведения за компьютером провоцирует появление компьютерной зависимости – пристрастия к занятиям, связанным с использованием компьютера, приводящее к резкому сокращению всех остальных видов деятельности. В статье описывается разработка мобильного приложения, позволяющего диагностировать, а в дальнейшем бороться с компьютерной зависимостью. Приложение разработано на языке программирования JAVA и предназначено для работы на любом Android-устройстве.

*Ключевые слова:* мобильное приложение, программирование, JAVA, Android, компьютерная зависимость.

## 1. Введение

Одним из самых значимых изобретений середины XX века было создание электронно-вычислительной машины, которая позднее в индивидуальном варианте получила название персональный компьютер (ПК). Развитие этой технологии привело к появлению мобильных компьютеров: ноутбуков, планшетов и смартфонов, которые позволяют пользоваться почти теми же ресурсами что и при работе за ПК, при этом находясь практически в любой точке Земли. Сегодня мобильные устройства подчас на равных конкурируют со стационарными ПК, поэтому в данной работе мы для удобства будем называть все эти устройства просто компьютерами. Широкое распространение разнообразной компьютерной техники со значительно возросшей функциональностью сделало её неотъемлемой частью жизни большинства людей особенно в развитых странах, к которым можно отнести и Россию.

Такое массовое распространение компьютеров, сопровождающееся увеличением всевозможного программного обеспечения и получаемой информации, не могло пройти бесследно. Люди стали проводить за ними очень много времени. Для понимания масштаба проблемы укажем, что в мире на начало 2021 года 5,22 млрд человек (66% населения) пользуются мобильными телефонами, а количество интернет-пользователей выросло до 4,66 млрд (59,5%). А пользователей социальных сетей сейчас в мире насчитывается 4,2 млрд (53,6%). Если тенденция сохранится, то в 2021 году все пользователи в мире проведут в соцсетях в общей сложности 3,7 триллиона часов, что эквивалентно более 420 миллионам лет. Россияне проводят в соцсетях 2 часа 28 минут, что близко к среднемировому значению 2 часа 25 минут [1].

Огромное количество информации и времени проведения за компьютером провоцирует появление компьютерной зависимости – пристрастия к занятиям, связанным с использованием компьютера, приводящее к резкому сокращению всех остальных видов деятельности. Всемирная организация здравоохранения (ВОЗ) внесла в Международную классификацию болезней 11-го пересмотра (МКБ-11) зависимость от компьютерных игр под названием «игровое расстройство» (6C51, 6C51.0, 6C51.1). Теперь чрезмерное пристрастие к видеоиграм рассматривается, как заболевание.

Основная проблема состоит в том, что не все могут оградить работу за компьютером от развлечений, и порой вместо поиска полезной информации, тратят время на развлечения, начиная проводить за ним все больше времени. Причём от этого страдает не только работа, состояние организма также однозначно ухудшается при постоянном использовании компьютеров не только на работе, но и в свободное время. Данная проблема имеет место для всех категорий людей, как детей, так и взрослых. В том числе и для сотрудников, работающих в сфере железнодорожного транспорта, неразрывно связанных с выполнением должностных обязанностей на ПК.

Основной целью данной работы была разработка приложения для мобильных устройств, которое способно оказать помощь в диагностике компьютерной зависимости с помощью тестирования, а также в её лечении и профилактике. Данное приложение должно быть универсально и подойти для любого предприятия, любой организации и каждому человеку.

## 2. Выявление компьютерной зависимости и её признаки

Для реализации данной идеи необходимо проанализировать информацию о компьютерной зависимости, определить причины, способы диагностики, лечения и профилактики зависимости; реализовать систему тестирования и анализа, которая сразу же определит степень зависимости; сформировать на основании теста программу лечения и профилактики зависимости.

В качестве основных причин зависимости можно выделить следующие факторы [2, 3]:

1. Проблемы с общением и социальной адаптацией, поскольку человек, которому трудно завести друзей в реальной жизни, боящийся отвержения и критики, увлекается виртуальным общением, дающем человеку возможность создать любой образ и остаться анонимным. Это раскрепощает, снимает неуверенность в себе, что в итоге приводит к компенсации или замене реального общения онлайн-общением.

2. Нехватка поддержки и одобрения, провоцирующая постоянную проверку мессенджеров, электронной почты, соцсетей. Человек, получая очередное сообщение, чувствует личную значимость. И боится, что, если не ответит вовремя, его перестанут принимать в обществе.

3. Недостаток эмоций и нехватку собственных впечатлений компенсируют игры, фильмы, чужие фотографии, видео. Для ярких событий в собственной жизни нужна активность. Куда проще является чужих приключений.

Замкнутость и неуверенность в себе - так можно обобщить все вышеперечисленные причины и факторы, ведь человек, боящийся общения, считающий, что ничего не достиг и остался без поддержки, впоследствии становится замкнутым и неуверенным в себе, пытающимся реализоваться, избегая реальности.

Несомненно, необходимо бороться с причинами зависимости, но для того, чтобы начать борьбу необходимо диагностировать наличие зависимости. Основными ее признаками будут:

1. Сильное раздражение при попытках ограничить время игры или интернет-серфинга.
2. Повышение настроения в периоды, которые он проводит за компьютером.
3. Избегание личного общения, предпочтение переписки через интернет или социальные сети в отличие от живого общения.
4. Отказ выходить на улицу, отсутствие интереса где-либо, кроме игр или поиска чего-либо в сети.

Для сбора данных существует две формы, тестовые вопросы с предложенными вариантами ответов и вопросы с открытым ответом, где человек сам составляет свой ответ, для большей точности сбора и анализа используем первый вариант. Проанализировав ряд статей о выявлении и диагностике зависимости, наиболее точная и корректная система тест-опроса с четырьмя вариантами ответа, которые характеризуют периодичность, с которой происходит та или иная или ситуация, или действие в повседневной жизни человека, тест составлен таким образом, что «обмануть» его очень сложно, но все же тест создавался с задумкой того, что человек будет отвечать честно.

Таким образом, тест, использованный в мобильном приложении, состоит из 11 вопросов на каждый из которых можно дать ответ в виде оценки от 1 до 4. Тест составлен таким образом, что «обмануть» его сложно, но все же он создавался, исходя из того, что человек будет отвечать честно. В ответе нужно охарактеризовать частоту, с которой происходит та или иная ситуация, описанная в вопросе, в повседневной жизни человека. Варианты ответов:

1. Очень часто (+4)
2. Часто (+3)
3. Иногда (+2)
4. Редко (+1)

Примеры таких вопросов [4]:

Как часто Вы ощущаете оживление, удовольствие, удовлетворение или облегчение, находясь за компьютером (в сети)?

Как часто Вам удастся самостоятельно прекратить работу за компьютером (в сети)?

Как часто Вы пренебрегаете семейными, общественными обязанностями и учебой из-за частой работы за компьютером (пробывания в сети)?

Как часто Вы отмечаете нарушения сна или изменения режима сна в связи с частой работой за компьютером (в сети)?

И т.д.

По итогам опроса получаем сумму баллов всех ответов, на основе которой диагностируем степень зависимости или ее отсутствие. Всего выделяем 4 возможных результата:

До 15 баллов – «0 % риска развития компьютерной зависимости»;

16 – 22 балла – «Стадия увлеченности, появление потребности в компьютере»;

23 – 37 баллов – «Риск развития компьютерной зависимости (в последующем необходимо провести профилактические мероприятия)»;

Больше 38 баллов – «Наличие компьютерной зависимости!». Необходимо проводить лечение.

После диагностирования компьютерной зависимости можно приступать к её лечению, для чего существуют различные методы. Мы будем использовать методику «The 30-Day Digital Detox Challenge» («30-дневный курс цифровой детоксикации»), автором которой является психотерапевт Нэнси Колие (Colier) [5, 6]. Она изложила её в книге «Не могу перестать думать».

Для отказа от компьютерной зависимости она предлагает заменять вредные привычки более здоровыми и не связанными с компьютерными технологиями. Соответственно каждый день приложение выдаёт задание, которое заставляет вас сделать одну замену. Начать надо с перемещения компьютера в менее удобное место в доме. В первый день обращайте внимание и внутренне отмечайте каждый раз, когда вы чувствуете желание, или просто мелькает мысль проверить одно из своих мобильных устройств или компьютер. Когда вы это заметите, то спросите себя: «Я проверяю по привычке?» и «Нужна ли эта проверка прямо сейчас?» Если ответ «Привычка» или «Не нужно», то повторяйте про себя «Стоп!» и делайте именно так. Одновременно назначьте три раза в день, когда вам разрешено проверять свое устройство, независимо от необходимости. Затем надо отказаться от использования любых устройств при живом общении (на 2-й день), отказаться от ношения мобильного телефона в кармане (на 3-й день), превратить свою ванную комнату в зону, свободную от компьютеров (на 13-й день). Так вы добавляете каждый день по одному новому условию своего поведения, пока не начнете использовать все тридцать одновременно. Методика очень проста, но в то же время действенна.

На данном этапе разработки программы действия при профилактике не отличаются от лечения.

### **3. Инструменты разработки**

Для написания программы был выбран один из популярнейших языков программирования JAVA, он нетребователен к системным требованиям, поддерживает ООП и имеет множество библиотек с необходимыми функциями. Версия Android выбрана 4.1, чтобы приложение можно было запустить практически на любом Android-устройстве. В качестве среды разработки использовалась Android Studio [7, 8].

### **4. Дизайн мобильного приложения**

Мобильное приложение имеет интуитивно понятный графический интерфейс. Примеры окон приведены на рисунке. На данный момент имеется две темы оформления: светлая и тёмная.



Рис. 1. Примеры диалоговых и информационных окон приложения

Выполнение программы сопровождается релаксирующей музыкой, которая может быть отключена.

Приложение полностью переведено на 2 языка: русский и английский, язык меняется вместе со сменой языка устройства.

## Литература

1. Вся статистика интернета и соцсетей на 2021 год — цифры и тренды в мире и в России. URL: <https://www.web-canape.ru/business/vsya-statistika-interneta-i-socsetej-na-2021-god-cifry-i-trendy-v-mire-i-v-rossii/> (дата обращения: 17.04.2023).
2. Войскунский А.Е. Актуальные проблемы зависимости от интернет // Психологический журнал. 2014. № 1. С. 90–100.
3. Краснова С.В., Носатова М.О., Чапова О.Е., Быковская Е.В., Тундалева В.С., Казарян Н.Р. Как справиться с компьютерной зависимостью. Москва, Изд-во Эксмо, 2008. 224 с.
4. Юрьева Л.Н., Больбот Т.Ю. Компьютерная зависимость: формирование, диагностика, лечение и профилактика. Днепропетровск, Изд-во «Пороги», 2006. 196 с.
5. Colier N. Try The 30-Day Digital Detox Challenge // «Forbes», 04.01.2017. URL: <https://www.forbes.com/sites/nextavenue/2017/01/04/try-the-30-day-digital-detox-challenge/#138451823a3d> (дата обращения: 17.04.2023).
6. Как избавиться от цифровой зависимости за 30 дней. URL: <https://www.wrike.com/ru/blog/kak-izbavitsa-ot-tsifrovoy-zavisimosti-za-30-dney/> (дата обращения: 17.04.2023).
7. Дейтел П., Дейтел Х., Уолд А. Android для разработчиков. Изд-во «Питер», 2016. 512 с.
8. Пирская Л.В. Разработка мобильных приложений в среде Android Studio. Ростов-на-Дону, Изд-во ЮФУ, 2019. 123 с.

# Техника обучения нейронной сети по распознаванию объекта на фото с использованием 3D-модели

*В.С. Привезенов*

*Нижегородский Государственный Технический Университет НГТУ им. Р.Е. Алексеева*

Работа посвящена исследованию возможности использования 2D представлений 3D объектов для обучения нейронных сетей, которые решают задачи классификации и обнаружения объектов на фото. Реализована библиотека по генерации 2D представлений по 3D модели. Определены перспективы дальнейших исследований.

*Ключевые слова:* 3D модель, 2D отображения, набор данных, распознавание образов, нейронная сеть.

## 1. Введение

Для обучения нейронной сети для задачи классификации статических объектов необходимо собрать большой набор размеченных изображений. Многие компании нанимают команды людей для ручной разметки фотографий, на которых обучаются нейронные сети.

Ученые из Колумбийского университета разработали набор изображений COIL100, состоящий из 7200 цветных фотографий 100 различных объектов. Объекты размещали на вращающейся платформе на черном фоне и делали снимок каждые 5 градусов, тем самым получая 72 фотографии для каждого объекта. Этот процесс занимает много времени [1].

Еще одним примером трудоемкого проекта является библиотека Microsoft COCO с набором размеченных изображений, включающим 330 000 фотографий и более 1,5 миллиона объектов, все из которых были размечены вручную [2].

Для обучения нейронной сети, которая будет классифицировать объекты по фото мы воспользуемся 3D-моделью объекта [3]. Метод заключается в том, чтобы вращать 3D-модель с определенным шагом угла, и на каждой итерации делать снимок объекта с помощью виртуальной камеры. Для автоматизации этого процесса создадим библиотеку.

## 2. Работа библиотеки и тестирование

### 2.1 Работа библиотеки

Разработанная библиотека отображает нашу модель, вращает ее и делает снимок, далее (рис. 1) представлена съемка трех моделей: Эйфелевой башни, Пизанской башни и Венеры Милосской. Все изображения являются результатом работы библиотеки на разных углах поворота.

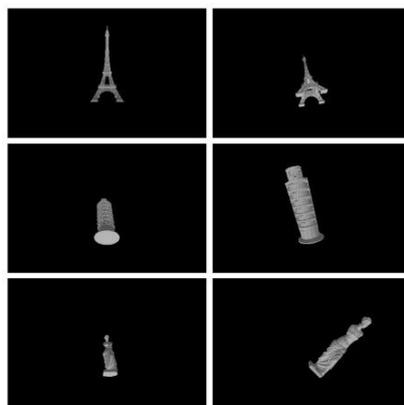
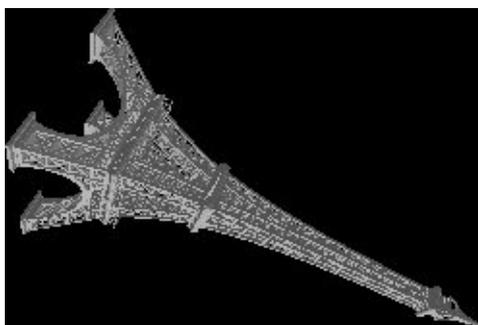


Рис. 1. Пример генерации изображений с разных ракурсов

Реализованная библиотека для получения 2D представлений 3D объекта обрабатывает полученные изображения следующим образом. На каждом полученном кадре сначала обрезаются черные края. Библиотека проверяет каждый пиксель и возвращает матрицу строк и столбцов, где на их пересечении имеются ненулевые пиксели (рис. 2).



**Рис. 2.** Изображение с обрезанными черными краями

После этого, с помощью алгоритма Отсу, находится оптимальное условие перевода цвета в черный или белый цвет и изображение переводится в бинарную цветовую гамму (рис. 3).



**Рис. 3.** Перевод изображения в бинарную цветовую гамму с помощью алгоритма Отсу

Далее, изображение растягивается до исходного размера (рис. 4), все это делается для того, чтобы каждое представление 3D объекта было нормализовано.



**Рис. 4.** Растягивание изображения до исходного разрешения

Полученные изображения выглядят следующим образом (рис. 5).



**Рис. 5.** Пример подготовленных изображений

Их действительно много (рис. 6), количество получаемых изображений устанавливает пользователь, все остальное библиотека сделает самостоятельно.

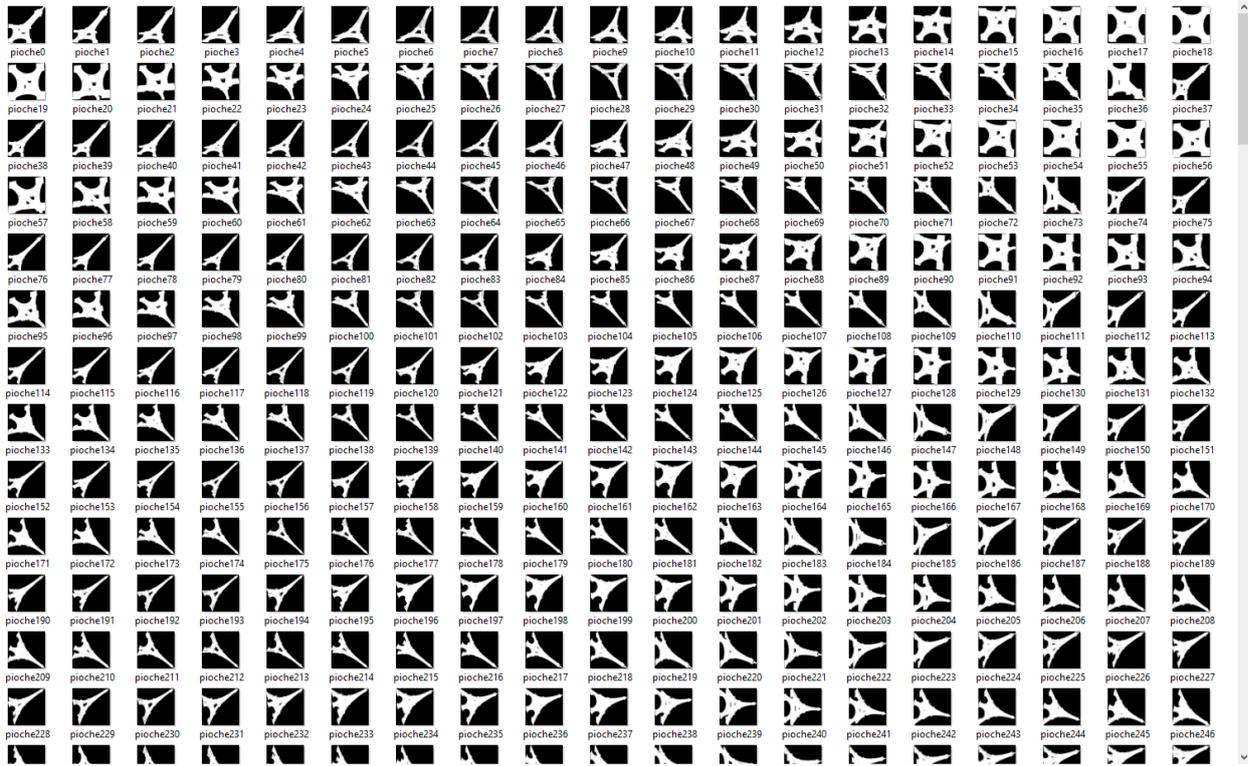


Рис. 6. Общее количество полученных изображений

## 2.2 Тестирование библиотеки на модели сверточной нейронной сети VGG16

Для обучения сверточной нейронной сети VGG16 [4] воспользуемся изображениями, сгенерированными нашей библиотекой. Объекты для генерации изображений принадлежат трем категориям: статуя Венеры Милосской, Пизанская башня, Эйфелева башня. Для тестирования возьмем реальные изображения, предварительно их подготовив с помощью библиотеки OpenCV (рис. 7).

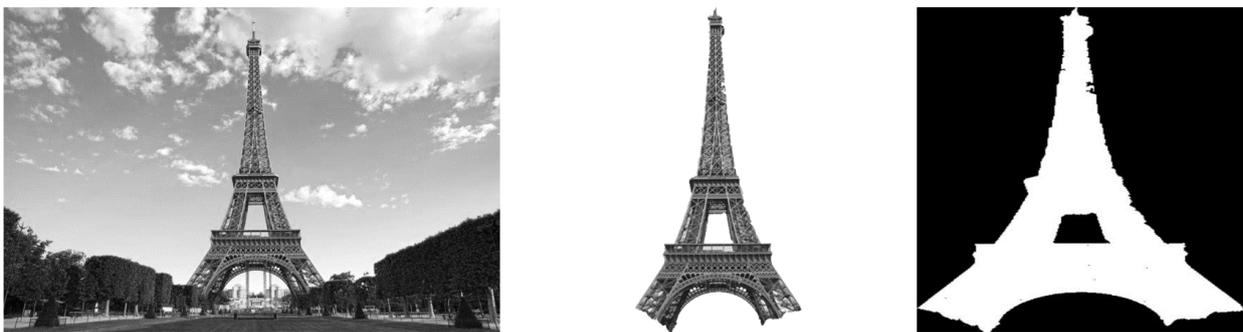


Рис. 7. Предварительная обработка изображений с помощью библиотеки OpenCV

Результаты тестирования оказались следующими (рис. 8): все изображения были распознаны верно, кроме последнего. Вероятно, последнее изображение было неверно распознано из-за его искаженности, ведь снимок сделан с необычного ракурса. Решение этой проблемы может заключаться в добавлении в обучающую выборку обрезанных изображений.



Рис. 8. Результаты тестирования обученной модели

## 2.3 Тестирование библиотеки на модели нейронной сети YOLOv5

Для обучения нейронной сети YOLOv5 воспользуемся сервисом [5], который автоматизирует работу с выборками и обучением. В качестве обучающих и валидационных изображений возьмем отображения, сгенерированные нашей библиотекой, по трем категориям: статуя Венеры Милосской, Пизанская башня, Эйфелева башня.

```
# train yolov5s on custom data for 100 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 100 --data './data.yaml' --cfg ./models/custom_yolov5s.yaml --weights '' --name yolov5s_results --cache
```

Epoch	gpu_mem	box	obj	cls	total	targets	img_size
0/99	1.41G	0.06189	0.02557	0.03027	0.1177	42	416: 100% 242/242 [00:35<00:00, 6.77it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:02<00:00, 2.98it/s]
all		184	0		0	0	0
Epoch	gpu_mem	box	obj	cls	total	targets	img_size
1/99	1.39G	0.05568	0.02592	0.03035	0.112	43	416: 100% 242/242 [00:32<00:00, 7.49it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:01<00:00, 5.63it/s]
all		184	0		0	0	0
Epoch	gpu_mem	box	obj	cls	total	targets	img_size
2/99	1.39G	0.05284	0.02604	0.02888	0.1078	42	416: 100% 242/242 [00:31<00:00, 7.61it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:00<00:00, 6.03it/s]
all		184	184	0.706	0.0833	0.0219	0.00234
Epoch	gpu_mem	box	obj	cls	total	targets	img_size
3/99	1.39G	0.04437	0.02778	0.02706	0.09921	37	416: 100% 242/242 [00:31<00:00, 7.76it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:00<00:00, 6.09it/s]
all		184	184	0.0193	0.196	0.00898	0.000964
Epoch	gpu_mem	box	obj	cls	total	targets	img_size
4/99	1.39G	0.03662	0.02863	0.02486	0.09011	43	416: 100% 242/242 [00:30<00:00, 7.81it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:01<00:00, 5.93it/s]
all		184	184	0.447	0.492	0.158	0.0238
Epoch	gpu_mem	box	obj	cls	total	targets	img_size
5/99	1.39G	0.03533	0.02749	0.02397	0.08679	39	416: 100% 242/242 [00:31<00:00, 7.76it/s]
Class		Images	Targets		P	R	mAP@.5 mAP@.5:.95: 100% 6/6 [00:00<00:00, 6.02it/s]
all		184	184	0.136	0.559	0.157	0.0281

Рис. 9. Процесс обучения YOLOv5

Изменение метрик качества модели во время обучения представлена на рис. 10.

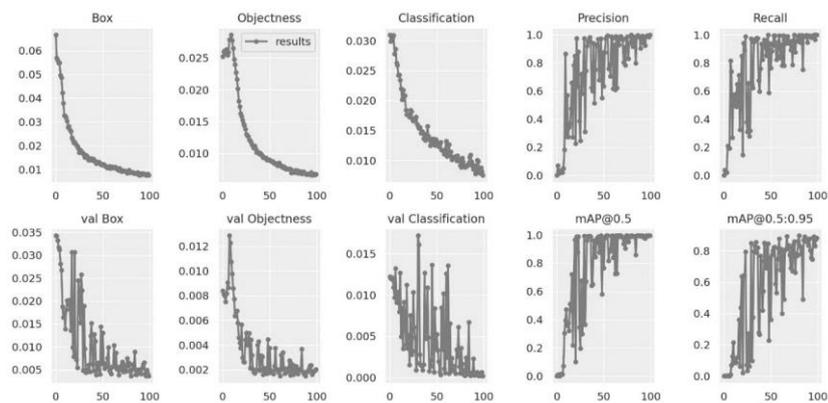


Рис. 10. Изменение показателей метрик во время обучения

Тестирование обученной модели проводилось сразу на реальных изображениях. Для наглядности, на объектах в рамке выведен процент определения с максимально подходящей категорией. Результат тестирования оказался следующим (рис. 11):



**Рис. 11.** Результат обученной модели YOLOv5

Не для всех тестовых изображений была верно определена категория, однако модели семейства YOLO имеют более сложную структуру. Так же, факт того, что часть изображений была определена верно, говорит о том, что идея использования генерации изображений из 3D модели для решения задачи обнаружения объектов имеет большие перспективы и должна развиваться дальше.

## 2.4 Результаты

Создана библиотека для генерации обучающей выборки изображений объекта из его 3D модели. Была обучена модель нейронной сети VGG16 на созданной выборке, обученная нейронная сеть показала способность классифицировать реальные фотографии. Была обучена модель нейронной сети YOLOv5 на созданной выборке, и обученная нейронная сеть показала способность распознавать объекты на реальных фотографиях. Таким образом, было продемонстрировано, что техника действительно позволяет обучать нейронные сети для распознавания объектов. Исходное предположение подтвердилось. Созданная библиотека может быть использована для обучения других нейронных сетей и позволит значительно упростить процесс создания обучающей выборки для широкого класса объектов. В дальнейшем планируется реализация графической оболочки, новые методы подготовки изображений для улучшения результатов, автоматические алгоритмы разметки изображений для наиболее популярных моделей нейронных сетей.

## Литература

1. Columbia University Image Library. (дата обращения: 10.04.2023)  
URL: <https://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>.
2. Common objects in context. URL: <https://cocodataset.org/> (дата обращения: 05.04.2023).
3. Multi-view Convolutional Neural Networks for 3D Shape Recognition. URL: <https://vis-www.cs.umass.edu/mvcnn/docs/su15mvcnn.pdf> (дата обращения: 05.04.2023).
4. VGG16 – Convolutional Network for Classification and Detection. URL: <https://github.com/ashushekar/VGG16> (дата обращения: 02.04.2023).
5. YOLOv5. URL: <https://roboflow.com/> (дата обращения: 04.04.2023).

# Применение глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL

*В.В. Разин, А.А. Краснов*

*Нижегородский государственный университет имени Н.И. Лобачевского*

Диагностика по электрокардиограмме (ЭКГ) является крайне актуальной и важной задачей, от качества, своевременности и скорости проведения которой зависят жизни и здоровье людей. В рамках статьи исследуется возможность применения различных методов глубокого обучения с различными входными данными как универсального инструмента для решения задачи при определении патологических сигналов ЭКГ с маркерами инфаркта миокарда, гипертрофии, нарушением проводимости и изменении морфологии ST сегмента. Производится сравнение использования двух отличных друг от друга подходов в обучении нейронных сетей и построении на их основе различных ансамблей с последующим анализом проведенных экспериментов.

*Ключевые слова:* глубокое обучение, диагностика ЭКГ, рекуррентная нейронная сеть, сверточная нейронная сеть, ансамбль, искусственный интеллект, multilabel classification, multiclass classification.

## 1. Введение

Широкое распространение во всем мире сердечно-сосудистых заболеваний (ССЗ) во второй половине XX века послужило поводом рассматривать их как «эпидемию ССЗ» [1]. ССЗ будут оставаться наиболее актуальной проблемой здравоохранения большинства стран мира в XXI веке [2]. Смертность от сердечно-сосудистых заболеваний занимает одно из лидирующих мест в общей структуре смертности [3]. Заболевания сердечно-сосудистой системы по-настоящему можно считать главными в формировании здоровья нации. По статистике, сформированной всемирной организацией здравоохранения, процент смертей от заболеваний сердца и сосудов хоть и неуклонно уменьшается, но все еще остается непомерно высоким: на 2020 год, процент людей с различными патологическими состояниями сердечной мышцы составляет 31%. Большинство врачей кардиологов сходятся во мнении, что данный тип заболеваний проще купировать на ранних стадиях развития.

Среди патологий, вызывающих повышенное внимание у кардиологов, являются инфаркт миокарда, нарушение проводимости сердечной ткани, гипертрофия и изменение ST сегмента сигнала ЭКГ. Каждое из представленных заболеваний является критически опасным для человека и может свидетельствовать о нарушении питания сердечной мышцы, возникновении некроза, появлении сердечных тканей с нарушением проводимости нервных импульсов, увеличении рабочей нагрузки и прочее.

Последние годы наблюдается относительное снижение показателей смертности от сердечно-сосудистых заболеваний, но Россия по этому критерию находится на первом месте в сравнении со странами Евросоюза [4]. Стоит отметить, что с 2008 по 2018 г. смертность от ССЗ в России снизилась на 32% [5]. Улучшение качества медицинской помощи позволяют оказывать влияние на динамику смертности населения от сердечно-сосудистых заболеваний [4]. На сегодняшний день в российском здравоохранении активно интегрируется механизм всеобщего, комплексного скринингового исследования населения. Такой подход позволит выявить предикторы ранней деградации сердечно-сосудистой системы у пациента. Однако, в следствии критически большой несбалансированности числа врачей и пациентов, реализовать полный контроль состояния каждого исследуемого невозможно. Одним из решений данной проблемы состоит в применении систем поддержки принятия решений, базирующихся на комбинации диагностических возможностей математических алгоритмов и методов машинного обучения.

Также за последние несколько лет совершен значительный прогресс в сфере автоматизации диагностики по ЭКГ, на ряде подзадач компьютер уже превзошел кардиологов в их решении [6],

а также проводится большое количество исследований в этой сфере [7]. Благодаря накопленным за многие годы кардиологическим данным, сегодня исследователи могут использовать алгоритмы машинного обучения в полную силу, анализируя как качество исходных данных, так и возможности применяемых методов. Важно и то, что данные являются свободно распространяемыми и обезличенными, что позволяет без ограничений исследовать исходный сигнал.

Наиболее востребованными для анализа сигналов являются различные нейронные сети, показавшие отличные результаты при обработке двумерных сигналов, в частности изображений и являющихся неоптимальными, но, практически всегда, рабочими решениями, в том числе в анализе биологически активных сигналов. В рамках статьи рассматривается анализ двенадцатиканальных ЭКГ, полученных с десяти отведений кардиографа, с целью определения нормальных кардиограмм (не имеющих явных отклонений от нормы), а также сигналов с маркерами нарушения проводимости, гипертрофии, инфаркта и элевации ST сегмента.

Обычно наборы данных ЭКГ очень малы [8], но не в случае набора данных РТВ-XL [9], размещенном PhysioNet [10]. В рамках данной работы на этом наборе данных решается задача диагностики электрокардиограммы классификации основных подклассов диагнозов. В качестве инструмента анализа ЭКГ используется глубокое обучение, демонстрирующее хорошие показатели решения задач классификации в их различном многообразии, а также в задачах диагностики ЭКГ [11-15].

## 2. Данные и методы

### 2.1 Набор данных РТВ-XL

В данном разделе кратко представляется набор данных РТВ-XL [9], который лежит в основе экспериментов, представленных ниже. Набор данных РТВ-XL содержит 21837 клинических 12-ти канальных записей ЭКГ длительностью 10 секунд с частотой замера 100 Гц. Также имеются диагностические утверждения, состоящие из 5 подклассов (NORM: нормальная ЭКГ, CD: нарушение проводимости, MI: инфаркт миокарда, HYP: гипертрофия и STTC: изменения ST/T). Дополнительная и вся основная информация о наборе данных содержится в оригинальной публикации [9].

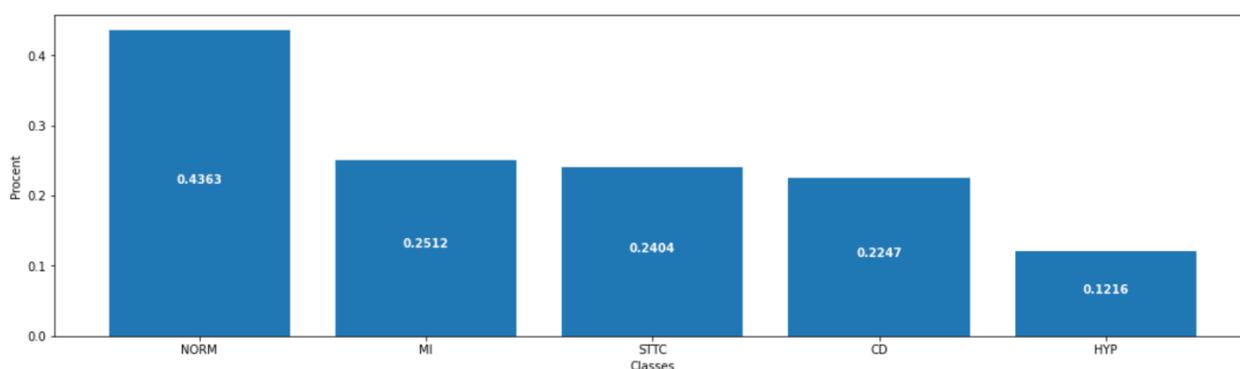


Рис. 1. Распределение классов в наборе данных РТВ XL

### 2.2 Алгоритмы классификации временных рядов

Для сравнительного анализа различных алгоритмов классификации данная работа основана на алгоритмах, которые работают с необработанными многомерными данными временных рядов. Подходы глубокого обучения для классификации временных рядов описаны во множестве обзоров [16-19].

На основе анализа применимости разных архитектур [20-22] в этой работе используются 3 основных категории архитектур глубоких нейронных сетей:

- сверточные нейронные сети;
- рекуррентные нейронные сети;
- комбинация сверточной и рекуррентной сетей.

Подробно посмотреть архитектуры всех сетей можно в публичном Notebook на Kaggle [23]. Модели были обучены на исходных данных временных рядов без какой-либо дополнительной предварительной обработки или с использованием нормализации средним.

## 2.3 Метрики классификации с несколькими метками

В качестве метрик оценки моделей классификации было отдано предпочтение метрике ABS (процент абсолютно точно предсказанных диагнозов), которая наглядно показывает процент точных предсказаний конкретной модели на основании статьи с такой же задачей, но с применением машинного обучения [24]. Данная метрика показывает абсолютную точность предсказаний и хорошо распределяет различные методы машинного обучения по их способности предугадывать нужный набор подклассов.

## 3. Результаты экспериментов

### 3.1 Описание задачи

Решается задача классификации с несколькими метками предсказания наличия подклассов (NORM, MI, HYP, STTC, CD) по входной 12-канальной 10 секундной записи ЭКГ с частотой замера 100 Гц. В одной из предыдущих работ [25] по итогам экспериментов были взяты 10 лучших архитектур, которые продемонстрировали наибольшую предсказательную способность по сравнению с другими. Предлагаемая оценка по ABS применяется одинаково для каждого эксперимента.

По итогу в эту десятку лучших попали:

- обычная сверточная сеть [26] (далее CNN)
- глубоко разделяемая сверточная сеть [27] (далее SCNN)
- сеть основанная на AlexNet [28] (далее AlexNet)
- комбинация сверточной сети с двунаправленными [29] LSTM и GRU (далее CBV)
- сеть основанная на Inception [30] (далее Inception)
- рекуррентная сеть LSTM [21] (далее LSTM)
- рекуррентная сеть GRU [22] (далее GRU)
- комбинация LSTM и GRU (далее LG)
- сеть основанная на VGG16 [31] (далее VGG16)
- сеть основанная на Xception [32] (далее Xception)

### 3.2 Ансамбль с лучшей предсказательной способностью

По итогам экспериментов был найден ансамбль, состоящий из Inception, CBV, LSTM и LG, демонстрирующий наибольший показатель ABS равный 71.21%.

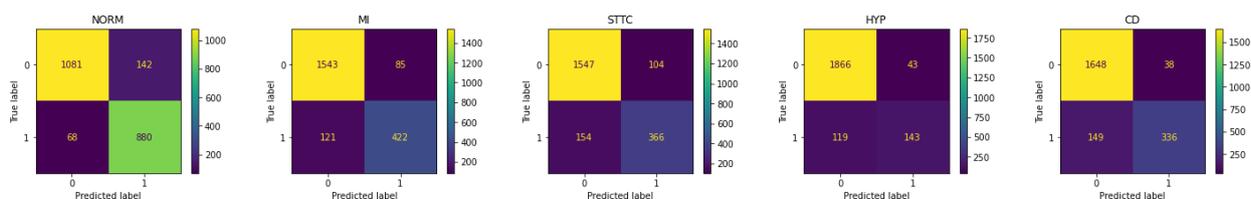


Рис. 2. Матрица ошибок лучшего ансамбля (по ABS)

### 3.3 Детали экспериментов

Архитектуры, деление на обучающую и проверочную выборки, использование нормализации средним, сам процесс обучения и все остальное заимствованы из одной из предыдущих работ [25], в которой диагнозы распределены не вектором длиной 5, где 1 в позиции означает наличие конкретного класса, а 0 его отсутствие (далее multilabel). Однако в данной работе дополнительно рассматривается еще альтернативный подход, где диагноз представляет собой вектор длиной 32, где 1 означает наличие сразу множества диагнозов и только них, а 0 отсутствие данного конкретного множества (далее multiclass).

Обучение производилось одинаковое время (около 100 часов) на одной и той же видеокарте для каждого подхода (их набора нейронных сетей).

В начале демонстрируется сравнение двух вышеописанных подходов глубокого обучения. В таблице 1 демонстрируются показатели ABS для каждой сети и для каждого подхода. Наибольший показатель ABS для каждого случая выделен жирным шрифтом.

**Таблица 1.** Результаты экспериментов с отличным друг от друга подходов глубокого обучения нейронных сетей

Название сети	ABS	
	multilabel	multiclass
CNN	67.49	68.82
SCNN	67.13	68.68
AlexNet	67.58	69.18
CBB	<b>68.11</b>	<b>70.06</b>
Inception	66.10	68.08
LSTM	67.23	68.45
GRU	66.64	68.40
LG	67.36	69.74
VGG16	67.73	69.14
Xception	66.65	69.05

Можно заметить, что лучше всех по ABS выступают сети CBB, а также что каждая сеть по методу multiclass превосходит аналогичную по multilabel.

Далее происходит построение всех возможных ансамблей (по участникам) трех типов, которые применялись в одной из предыдущих работ [24] как для сетей, обученных по методу multilabel, так и для multiclass (но не вместе). Названия ансамблей также заимствованы из одной из предыдущих работ [24]. В Таблице 2 демонстрируются показатели ABS для каждого типа ансамбля и для каждого подхода. Наибольший показатель ABS для каждого случая выделен жирным шрифтом.

**Таблица 2.** Результаты экспериментов с отличным друг от друга подходами глубокого обучения ансамблей из соответствующих нейронных сетей

Название ансамбля	ABS	
	multilabel	multiclass
AE	69.09	70.94
KE	69.09	70.94
DE	<b>71.21</b>	<b>71.21</b>

Можно заметить, что лучше всех по ABS выступают ансамбли DE, а также то, что показатели данных ансамблей абсолютно идентичны. В состав обоих из них вошли модели Inception, CBV, LSTM и LG.

Более подробно ознакомиться со всем кодом можно в публичном Notebook на Kaggle.

#### 4. Заключение и выводы

Методы глубокого обучения позволяют достичь впечатляющей способности диагностики по ЭКГ на таком большом наборе данных как РТВ-XL. Подход multiclass демонстрирует лучшие показатели при обучении каждой рассмотренной архитектуры нейронной сети, однако используя ансамбли multilabel и multiclass демонстрируют абсолютно одинаковый показатель ABS для лучшего типа ансамбля DE. Ансамбль DE показывает лучшую предсказательную способность вне зависимости от выбранного метода обучения. Веса обученных моделей можно использовать для обучения на меньших наборах данных путем переноса обучения (как в случае работы с изображениями с переносом обучения с ImageNet [33]), а также из данных нейронных сетей извлекать необходимые признаки для моделей машинного обучения.

#### Литература

1. Шальнова С. А., Деев А. Д., Оганов Р. Г. Факторы, влияющие на смертность от сердечно-сосудистых заболеваний в российской популяции // Кардиоваскулярная терапия и профилактика. – 2005. – Т. 4. – № 1. – С. 4–9.
2. Оганов Р. Г., Погосова Г. В. Современные стратегии профилактики и лечения сердечно-сосудистых заболеваний // Кардиология. – 2007. – Т. 47. – № 12. – С. 4–9.
3. Murray C. J. L., Lopez A. D. Global mortality, disability, and the contribution of risk factors: Global Burden of Disease Study // The lancet. – 1997. – Т. 349. – № 9063. – С. 1436–1442.
4. Безверхов А. А., Ищенко О. Ю. ДИНАМИКА СМЕРТНОСТИ ОТ СЕРДЕЧНО-СОСУДИСТЫХ ЗАБОЛЕВАНИЙ ЗА ПЕРИОД 2015–2019 гг //Norwegian Journal of development of the International Science. – 2021. – № 54–2. – С. 35–38.
5. Бойцов С. А., Драпкина О. М. Современное содержание и совершенствование стратегии высокого сердечно-сосудистого риска в снижении смертности от сердечно-сосудистых заболеваний //Терапевтический архив. – 2021. – Т. 93. – № 1. – С. 4–6.
6. Attia Z. I. et al. Age and sex estimation using artificial intelligence from standard 12-lead ECGs //Circulation: Arrhythmia and Electrophysiology. – 2019. – Т. 12. – № 9. – С. e007284.
7. Hong S. et al. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review //Computers in Biology and Medicine. – 2020. – Т. 122. – С. 103801.
8. Schläpfer J., Wellens H. J. Computer-interpreted electrocardiograms: benefits and limitations //Journal of the American College of Cardiology. – 2017. – Т. 70. – № 9. – С. 1183–1192.

9. Wagner P. et al. PTB-XL, a large publicly available electrocardiography dataset //Scientific data. – 2020. – Т. 7. – № 1. – С. 1–15.
10. Goldberger A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals //circulation. – 2000. – Т. 101. – № 23. – С. e215–e220.
11. Attia Z. I. et al. Novel bloodless potassium determination using a signal-processed single-lead ECG //Journal of the American heart Association. – 2016. – Т. 5. – № 1. – С. e002746.
12. Hannun A.Y. et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network // Nature medicine, 2019 – Т. 25. – № 1. – С. 65–69.
13. Attia Z. I. et al. An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction //The Lancet, 2019. – Т. 394. – № 10201. – С. 861–867.
14. Attia Z. I. et al. Screening for cardiac contractile dysfunction using an artificial intelligence-enabled electrocardiogram //Nature medicine. – 2019. – Т. 25. – № 1. – С. 70–74.
15. Ribeiro A. H. et al. Automatic diagnosis of the 12-lead ECG using a deep neural network //Nature communications. – 2020. – Т. 11. – № 1. – С. 1–9.
16. Gamboa J. C. B. Deep learning for time-series analysis //arXiv preprint arXiv:1701.01887. – 2017.
17. Ismail Fawaz H. et al. Deep learning for time series classification: a review //Data mining and knowledge discovery. – 2019. – Т. 33. – № 4. – С. 917–963.
18. Roy Y. et al. Deep learning-based electroencephalography analysis: a systematic review //Journal of neural engineering. – 2019. – Т. 16. – № 5. – С. 051001.
19. Bagnall A. et al. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances // Data mining and knowledge discovery, 2017. Т. 31. № 3. С. 606–660.
20. Wang Z., Yan W., Oates T. Time series classification from scratch with deep neural networks: A strong baseline //2017 International joint conference on neural networks (IJCNN). IEEE, 2017. С. 1578–1585.
21. Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – № 8. – С. 1735–1780.
22. Cho K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation //arXiv preprint arXiv:1406.1078. – 2014.
23. Razin V., Krasnov A. AI PTBXL [Электронный ресурс] – URL: <https://www.kaggle.com/razinw/ai-ptbxl> (дата обращения 03.03.2023).
24. Краснов А.А., Разин В.В., Золотых Н.Ю. Машинное обучение в задачах диагностики заболевания по ЭКГ на наборе данных PTB-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С.44–50.
25. Разин В.В., Краснов А.А., Золотых Н.Ю. Глубокое обучение в задачах диагностики заболевания по ЭКГ на наборе данных PTB-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С.78–86.
26. Albawi S., Mohammed T. A., Al-Zawi S. Understanding of a convolutional neural network // 2017 international conference on engineering and technology (ICET). – Ieee, 2017. – С. 1–6.
27. Hu Z. et al. 3D separable convolutional neural network for dynamic hand gesture recognition //Neurocomputing. – 2018. – Т. 318. – С. 151–161.
28. Ismail Fawaz H. et al. Inceptiontime: Finding alexnet for time series classification //Data Mining and Knowledge Discovery. – 2020. – Т. 34. – № 6. – С. 1936–1962.

29. Huang Z., Xu W., Yu K. Bidirectional LSTM-CRF models for sequence tagging // arXiv preprint arXiv:1508.01991. – 2015.
30. Wang C. et al. Pulmonary image classification based on inception-v3 transfer learning model //IEEE Access. – 2019. – T. 7. – C. 146533–146541.
31. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556. – 2014.
32. Jinsakul N. et al. Enhancement of deep learning in image classification performance using xception with the swish activation function for colorectal polyp preliminary screening //Mathematics. – 2019. – T. 7. – № 12. – C. 1170.
33. Deng J. et al. Imagenet: A large-scale hierarchical image database //2009 IEEE conference on computer vision and pattern recognition. – Ieee, 2009. – C. 248–255.

# Применение машинного обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL

*В.В. Разин, А.А. Краснов*

*Нижегородский государственный университет имени Н.И. Лобачевского*

Диагностика по электрокардиограмме (ЭКГ) является крайне актуальной и важной задачей, от качества, своевременности и скорости проведения которой зависят жизни и здоровье людей. В рамках статьи исследуется возможность применения различных методов машинного обучения с различными входными данными как универсального инструмента для решения задачи при определении патологических сигналов ЭКГ с маркерами инфаркта миокарда, гипертрофии, нарушением проводимости и изменении морфологии ST сегмента. Производится сравнение использования признаков из двух отличных друг от друга подходов в обучении нейронных сетей в методах машинного обучения и построении на их основе различных ансамблей с последующим анализом проведенных экспериментов.

*Ключевые слова:* машинное обучение, ЭКГ, ансамбль, искусственный интеллект, признаки нейронных сетей, multilabel classification, multiclass classification.

## 1. Введение

Широкое распространение во всем мире сердечно-сосудистых заболеваний (ССЗ) во второй половине XX века послужило поводом рассматривать их как «эпидемию ССЗ» [1]. ССЗ будут оставаться наиболее актуальной проблемой здравоохранения большинства стран мира в XXI веке [2]. Смертность от сердечно-сосудистых заболеваний занимает одно из лидирующих мест в общей структуре смертности [3]. Заболевания сердечно-сосудистой системы по-настоящему можно считать главными в формировании здоровья нации. По статистике, сформированной всемирной организацией здравоохранения, процент смертей от заболеваний сердца и сосудов хоть и неуклонно уменьшается, но все еще остается непомерно высоким: на 2020 год, процент людей с различными патологическими состояниями сердечной мышцы составляет 31%. Большинство врачей кардиологов сходятся во мнении, что данный тип заболеваний проще купировать на ранних стадиях развития.

Среди патологий, вызывающих повышенное внимание у кардиологов, являются инфаркт миокарда, нарушение проводимости сердечной ткани, гипертрофия и изменение ST сегмента сигнала ЭКГ. Каждое из представленных заболеваний является критически опасным для человека и может свидетельствовать о нарушении питания сердечной мышцы, возникновении некроза, появлении сердечных тканей с нарушением проводимости нервных импульсов, увеличение рабочей нагрузки и прочее.

Последние годы наблюдается относительное снижение показателей смертности от сердечно-сосудистых заболеваний, но Россия по этому критерию находится на первом месте в сравнении со странами Евросоюза [4]. Стоит отметить, что с 2008 по 2018 г. смертность от ССЗ в России снизилась на 32% [5]. Улучшение качества медицинской помощи позволяют оказывать влияние на динамику смертности населения от сердечно-сосудистых заболеваний [4]. На сегодняшний день в российском здравоохранении активно интегрируется механизм всеобщего, комплексного скринингового исследования населения. Такой подход позволит выявить предикторы ранней деградации сердечно-сосудистой системы у пациента. Однако, в следствии критически большой несбалансированности числа врачей и пациентов, реализовать полный контроль состояния каждого исследуемого невозможно. Одним из решений данной проблемы состоит в применении систем поддержки принятия решений, базирующихся на комбинации диагностических возможностей математических алгоритмов и методов машинного обучения.

Также за последние несколько лет совершен значительный прогресс в сфере автоматизации диагностики по ЭКГ, на ряде подзадач компьютер уже превзошел кардиологов в их решении [6],

а также проводится большое количество исследований в этой сфере [7]. Благодаря накопленным за многие годы кардиологическим данным, сегодня исследователи могут использовать алгоритмы машинного обучения в полную силу, анализируя как качество исходных данных, так и возможности применяемых методов. Важно и то, что данные являются свободно распространяемыми и обезличенными, что позволяет без ограничений исследовать исходный сигнал.

Наиболее востребованными для анализа сигналов являются различные нейронные сети, показавшие отличные результаты при обработке двумерных сигналов, в частности изображений и являющихся неоптимальными, но, практически всегда, рабочими решениями, в том числе в анализе биологически активных сигналов. В рамках статьи рассматривается анализ двенадцатиканальных ЭКГ, полученных с десяти отведений кардиографа, с целью определения нормальных кардиограмм (не имеющих явных отклонений от нормы), а также сигналов с маркерами нарушения проводимости, гипертрофии, инфаркта и элевации ST сегмента.

Обычно наборы данных ЭКГ очень малы [8], но не в случае набора данных РТВ-XL [9], размещенном PhysioNet [10]. В рамках данной работы на этом наборе данных решается задача диагностики электрокардиограммы классификации основных подклассов диагнозов. В качестве инструмента анализа ЭКГ используется машинное обучение, демонстрирующее хорошие показатели решения задач классификации в их различном многообразии, а также в задачах диагностики ЭКГ [11-15] с высокой объяснимостью работы самих методов.

## 2. Данные и методы

### 2.1 Набор данных РТВ-XL

В данном разделе кратко представляется набор данных РТВ-XL [9], который лежит в основе экспериментов, представленных ниже. Набор данных РТВ-XL содержит 21837 клинических 12-ти канальных записей ЭКГ длительностью 10 секунд с частотой замера 100 Гц. Также имеются диагностические утверждения, состоящие из 5 подклассов (NORM: нормальная ЭКГ, CD: нарушение проводимости, MI: инфаркт миокарда, HYP: гипертрофия и STTC: изменения ST/T). Дополнительная и вся основная информация о наборе данных содержится в оригинальной публикации [9].

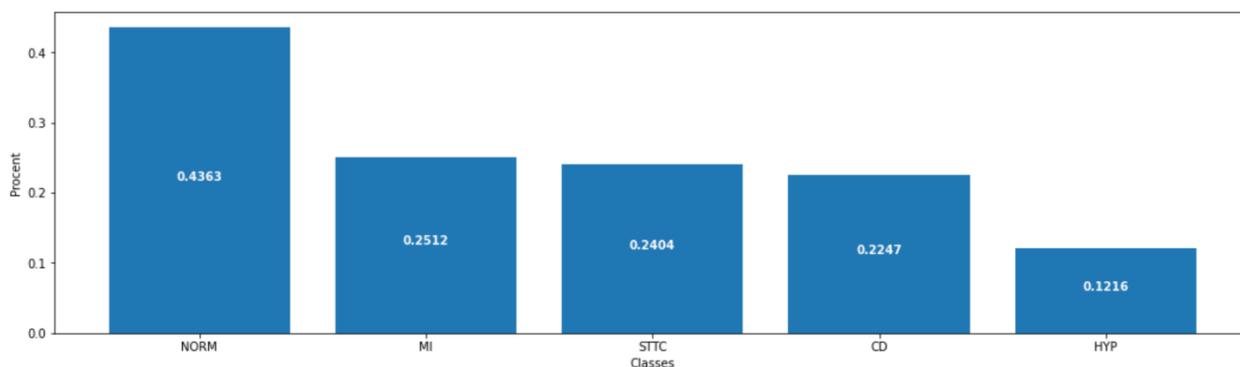


Рис. 1. Распределение классов в наборе данных РТВ XL

### 2.2 Алгоритмы классификации временных рядов

Для сравнительного анализа различных алгоритмов классификации данная работа основана на алгоритмах, которые работают с необработанными многомерными данными временных рядов. Подходы машинного обучения для классификации временных рядов описаны во множестве обзоров [16-17].

На основе анализа применимости разных моделей машинного обучения [18-22] в этой работе используются 10 различных методов машинного обучения.

Подробно посмотреть сами методы с их параметрами и результаты их предсказаний можно в публичном Notebook на Kaggle [23].

Модели были обучены на признаках, извлеченных из различных нейронных сетей (подробнее в разделе «Результаты экспериментов»), которые были обучены для решения этой же задачи, но с применением глубокого обучения [24].

### 2.3 Метрики классификации с несколькими метками

В качестве метрики оценки моделей классификации было отдано предпочтение метрике ABS (точность), которая наглядно показывает процент точных предсказаний конкретной модели на основании одной из предыдущих работ [25]. Данная метрика показывает абсолютную точность предсказаний и хорошо распределяет различные методы машинного обучения по их способности предугадывать нужный набор подклассов.

## 3. Результаты экспериментов

### 3.1 Описание задачи

Решается задача классификации с несколькими метками предсказания наличия подклассов (NORM, MI, HYP, STTC, CD) по признакам, извлеченным из различных нейронных сетей (архитектуры которых основаны на AlexNet [26], VGG [27] и глубоко разделяемой сверточной сети [28] (далее SCNN), обычной сверточной сети [29] (далее CNN), LSTM [30], GRU [31], комбинации LSTM и GRU (далее LG), Xception [32], Inception [33], комбинация сверточной сети с двунаправленными LSTM и GRU (далее CBV)), которые были обучены для решения этой же задачи [25].

В одной из предыдущих работ [25] по итогам экспериментов были взяты 10 лучших архитектур, которые продемонстрировали наибольшую предсказательную способность по сравнению с другими. Предлагаемая оценка по ABS применяется одинаково для каждого эксперимента.

По итогу в эту десятку лучших попали:

- стохастический градиентный спуск [18] (далее SGD)
- метод k-ближайших соседей [19] (далее KNN)
- пассивно-агрессивный классификатор [34] (далее PAC)
- линейный дискриминантный анализ [35] (далее LDA)
- дерево решений [21] (далее DT)
- классификатор экстремально рандомизированного дерева [36] (далее ET)
- классификатор экстремально рандомизированных деревьев [22] (далее ETS)
- случайный лес [20] (далее RF)
- регрессор Ridge [37] (RC)
- регрессор RidgeCV [38] (далее RCC)

### 3.2 Ансамбль с наибольшей предсказательной способностью

По итогам экспериментов был найден ансамбль, состоящий из KNN, LDA, ET, ETS, и RF, демонстрирующий наибольший показатель ABS равный 70.94%.

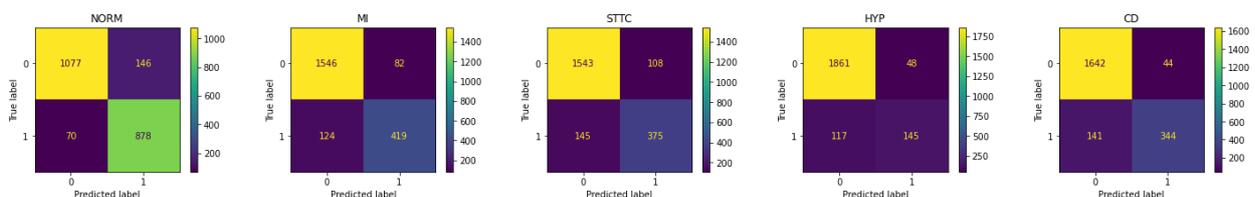


Рис. 2. Матрица ошибок лучшего ансамбля (по ABS)

### 3.3 Детали экспериментов

Гиперпараметры каждой модели машинного обучения, построение ансамблей и все остальное заимствованы из одной из предыдущих работ [25]. Однако в данной работе рассматривается взятие извлеченных признаков со всех нейронных сетей, обученных для решения такой же задачи методами глубокого обучения [24] и сравнение этих входных данных для каждого из подходов обучения (multilabel и multiclass) [24].

В начале методом полного перебора были определены лучшие комбинации извлеченных признаков для обоих методов обучения. Для multilabel лучшие показатели ABS демонстрирует комбинация из признаков из AlexNet, VGG16 и CNN, а для multiclass лучший ABS демонстрирует комбинация из признаков из VGG16, Inception, LSTM и LG.

Далее происходит сравнение показателей ABS для обучения на данных двух комбинациях признаков. В Таблице 1 демонстрируются показатели ABS для каждой метода машинного обучения и для каждой лучшей комбинации признаков. Наибольший показатель ABS для каждого случая выделен жирным шрифтом.

**Таблица 1.** Результаты экспериментов с различными комбинациями извлеченных признаков из нейронных сетей

Название модели	ABS	
	multilabel	multiclass
SGD	67.76	69.05
KNN	<b>70.29</b>	<b>70.29</b>
PAC	68.03	68.54
LDA	67.43	68.26
DT	60.43	59.01
ET	57.81	58.54
ETS	70.24	70.24
RF	69.09	70.20
RC	68.68	69.46
RCC	68.59	69.60

Можно заметить, что в обоих случаях лучше всех по ABS выступают KNN в обоих комбинациях признаков показывают одинаковый результат.

Далее происходит сравнение всех возможных ансамблей трех типов (названия и принцип построения заимствованы из одной из предыдущих работ [25]) для каждой входной комбинации признаков. В Таблице 2 демонстрируются показатели ABS для каждой типа ансамбля и для каждой лучшей комбинации признаков. Наибольший показатель ABS для каждого случая выделен жирным шрифтом.

**Таблица 2.** Результаты экспериментов с отличным друг от друга подходов глубокого обучения ансамблей из соответствующих нейронных сетей

Название ансамбля	ABS	
	multilabel	multiclass
AE	70.75	70.57
KE	70.75	70.57
DE	<b>70.94</b>	<b>70.75</b>

Можно заметить, что лучше всех по ABS выступают ансамбли DE, а также то, обучение на комбинации признаков, извлеченных из нейронных сетей, обученных по методу multilabel, и составление ансамбля пусть не сильно, но превосходит аналогичное по методу multiclass. В состав ансамбля DE, демонстрирующего наибольший показатель ABS равный 70.94%, вошли KNN, LDA, ET, ETS, и RF.

Более подробно ознакомиться со всем кодом можно в публичном Notebook на Kaggle.

#### 4. Заключение и выводы

Машинное обучение на признаках, извлеченных из нейронных сетей, может давать лучшие результаты предсказательной способности моделей, чем в самих нейронных сетях. Комбинация данных признаков также позволяет улучшить показатели методов машинного обучения. Комбинация, построенная на признаках сетей, демонстрирующих хуже показатели по методу multilabel в глубоком обучении, состоящая из меньшего количества признаков (по количеству сетей, с которых их извлекали) дает одинаковый результат на выбранных методах с заданными одинаковыми гиперпараметрами, а также превосходит по ABS метод обучения multiclass в случае ансамбля. Ансамбль DE показывает лучшую предсказательную способность вне зависимости от выбранного метода обучения. Полученная информация может быть полезна в дальнейших исследованиях при решении данной или аналогичных задач.

#### Литература

1. Шальнова С. А., Деев А. Д., Оганов Р. Г. Факторы, влияющие на смертность от сердечно-сосудистых заболеваний в российской популяции //Кардиоваскулярная терапия и профилактика. – 2005. – Т. 4. – № 1. – С. 4–9.
2. Оганов Р. Г., Погосова Г. В. Современные стратегии профилактики и лечения сердечно-сосудистых заболеваний //Кардиология. – 2007. – Т. 47. – № 12. – С. 4–9.
3. Murray C. J. L., Lopez A. D. Global mortality, disability, and the contribution of risk factors: Global Burden of Disease Study //The lancet. – 1997. – Т. 349. – № 9063. – С. 1436–1442.
4. Безверхов А.А., Ищенко О.Ю. Динамика смертности от сердечно-сосудистых заболеваний за период 2015–2019 г. //Norwegian Journal of development of the International Science. – 2021. – № 54–2. – С. 35–38.
5. Бойцов С. А., Драпкина О. М. Современное содержание и совершенствование стратегии высокого сердечно-сосудистого риска в снижении смертности от сердечно-сосудистых заболеваний //Терапевтический архив. – 2021. – Т. 93. – № 1. – С. 4–6.
6. Attia Z. I. et al. Age and sex estimation using artificial intelligence from standard 12-lead ECGs //Circulation: Arrhythmia and Electrophysiology. – 2019. – Т. 12. – № 9. – С. e007284.
7. Hong S. et al. Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review //Computers in Biology and Medicine. – 2020. – Т. 122. – С. 103801.

8. Schläpfer J., Wellens H. J. Computer-interpreted electrocardiograms: benefits and limitations //Journal of the American College of Cardiology. – 2017. – Т. 70. – № 9. – С. 1183–1192.
9. Wagner P. et al. PTB-XL, a large publicly available electrocardiography dataset //Scientific data. – 2020. – Т. 7. – № 1. – С. 1–15.
10. Goldberger A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals //circulation. – 2000. – Т. 101. – № 23. – С. e215–e220.
11. Attia Z. I. et al. Novel bloodless potassium determination using a signal-processed single-lead ECG //Journal of the American heart Association. – 2016. – Т. 5. – № 1. – С. e002746.
12. Hannun A. Y. et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network // Nature medicine. 2019. – Т. 25. – № 1. – С. 65–69.
13. Attia Z. I. et al. An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction //The Lancet. – 2019. – Т. 394. – № 10201. – С. 861–867.
14. Attia Z. I. et al. Screening for cardiac contractile dysfunction using an artificial intelligence-enabled electrocardiogram //Nature medicine. – 2019. – Т. 25. – № 1. – С. 70–74.
15. Ribeiro A. H. et al. Automatic diagnosis of the 12-lead ECG using a deep neural network //Nature communications. – 2020. – Т. 11. – № 1. – С. 1–9.
16. Sharma L. D., Sunkaria R. K. Inferior myocardial infarction detection using stationary wavelet transform and machine learning approach //Signal, Image and Video Processing. – 2018. – Т. 12. – № 2. – С. 199–206.
17. Oakden-Rayner L. et al. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging //Proceedings of the ACM conference on health, inference, and learning. – 2020. – С. 151–159.
18. Bottou L. Large-scale machine learning with stochastic gradient descent //Proceedings of COMPSTAT'2010. – Physica-Verlag HD, 2010. – С. 177–186.
19. Guo G. et al. KNN model-based approach in classification //OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". – Springer, Berlin, Heidelberg, 2003. – С. 986–996.
20. Liu Y., Wang Y., Zhang J. New machine learning algorithm: Random forest //International Conference on Information Computing and Applications. – Springer, Berlin, Heidelberg, 2012. – С. 246–252.
21. Charbuty B., Abdulazeez A. Classification based on decision tree algorithm for machine learning //Journal of Applied Science and Technology Trends. – 2021. – Т. 2. – № 01. – С. 20–28.
22. Geurts P., Ernst D., Wehenkel L. Extremely randomized trees //Machine learning. – 2006. – Т. 63. – № 1. – С. 3–42.
23. Razin V., Krasnov A. Machine learning [Электронный ресурс] – URL: <https://www.kaggle.com/razinw/machine-learning/> (дата обращения 08.03.2023).
24. Разин В.В., Краснов А.А. Применение глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ–XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов II Молодежной школы (Н. Новгород, 24–28 апреля 2023 г.). Нижний Новгород: Изд-во Нижегородского государственного университета, 2023. С.66–72.

25. Краснов А.А., Разин В.В., Золотых Н.Ю. Машинное обучение в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL // Математика и ИТ – вместе в цифровое будущее. Сборник трудов Молодежной школы (Н. Новгород, 25–29 апреля 2022 г.). Нижний Новгород: Изд-во Нижегородского госуниверситета, 2022. С.44–50.
26. Ismail Fawaz H. et al. Inceptiontime: Finding alexnet for time series classification //Data Mining and Knowledge Discovery. – 2020. – Т. 34. – № 6. – С. 1936–1962.
27. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.
28. Hu Z. et al. 3D separable convolutional neural network for dynamic hand gesture recognition //Neurocomputing. – 2018. – Т. 318. – С. 151–161.
29. Albawi S., Mohammed T. A., Al-Zawi S. Understanding of a convolutional neural network //2017 international conference on engineering and technology (ICET). – Ieee, 2017. – С. 1–6.
30. Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – № 8. – С. 1735–1780.
31. Cho K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation //arXiv preprint arXiv:1406.1078. – 2014.
32. Jinsakul N. et al. Enhancement of deep learning in image classification performance using xception with the swish activation function for colorectal polyp preliminary screening //Mathematics. – 2019. – Т. 7. – № 12. – С. 1170.
33. Wang C. et al. Pulmonary image classification based on inception-v3 transfer learning model //IEEE Access. – 2019. – Т. 7. – С. 146533–146541.
34. Crammer K. et al. Online passive aggressive algorithms. – 2006.
35. Tharwat A. et al. Linear discriminant analysis: A detailed tutorial //AI communications. – 2017. – Т. 30. – № 2. – С. 169–190.
36. Sharaff A., Gupta H. Extra-tree classifier with metaheuristics approach for email classification //Advances in computer communication and computational sciences. – Springer, Singapore, 2019. – С. 189–197.
37. Hoerl R. W. Ridge regression: a historical context //Technometrics. 2020. Т. 62. – № 4. – С. 420–425.
38. Yafoz A., Mouhoub M. Analyzing machine learning algorithms for sentiments in arabic text //2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2020. С. 2150–2156.

# Аугментация данных в обработке медицинских сигналов

*М.И. Самбурова*

*Нижегородский государственный университет имени Н.И. Лобачевского*

Возникающая в кардиологии задача обработки ЭКГ – это задача анализа одномерного сигнала. Рекуррентные нейронные сети хорошо справляются с задачами такого типа. В работе исследуется возможность применения архитектуры LSTM для классификации сердечных ударов в условиях нехватки данных пациентов с патологиями, а именно: влияние аугментации продвинутого типа (создание новых сигналов генеративно-состязательной сетью) на точность предсказания патологий.

*Ключевые слова:* машинное обучение, ЭКГ, глубокое обучение, генеративно-состязательные нейронные сети, рекуррентные нейронные сети, аугментация.

## 1. Введение

В современной науке о данных важное место занимает анализ временных рядов, последовательностей, одномерных сигналов. Данные такого типа встречаются совершенно в различных сферах, таких как банковское дело, социология, экономика, климатология, транспортная сфера и, наконец, медицина. В работе рассматривается задача обработки электрокардиограммы (ЭКГ). С задачами обработки ЭКГ хорошо справляются относительно новые инструменты – рекуррентные нейронные сети. Их особенность в том, что они анализируют участки последовательности с учетом соседней информации в этой последовательности. Такие инструменты обработки сигналов сердца могут вывести заботу о здоровье человека на новый уровень. Любые умные часы смогут отслеживать состояние пациента на уровне диагноза, а значит не будет необходимости в регулярных обследованиях с дорогой аппаратурой и профессионально обученным врачом, а постоянный мониторинг позволит сообщать о патологиях на самых ранних этапах.

Другой распространённой проблемой является недостаточный объем данных. ЭКГ каждого пациента нужно анализировать, учитывая сердечный ритм, частоту сокращений, а также множество характеристик присущих отдельным сердечным сокращениям. Для того, чтобы компьютер мог заменить врача, нужно проделать много работы: сбор данных, хранение, разметка, хранение информации о диагнозе, поставленном лечащим врачом.

Нейронные сети научились справляться и с проблемой нехватки данных, например, с помощью искусственной их генерации (аугментации). Есть простые варианты аугментации путём линейных преобразований, есть сложные – синтезировать принципиально новые объекты. Для этого хорошо подходят генеративные сети. В настоящее время генеративные модели способны заменить человека в некоторых областях труда. Уже даже вводятся законы, регулирующие работу ИИ.

В работе исследуется задача мультиклассовой классификации ударов сердца, применение улучшения рекуррентной нейронной сети (LSTM – Long Short Term Memory) для её решения, влияние дисбаланса на точность предсказания и предлагается вариант устранения этого дисбаланса с помощью DCGAN – Deep Convolutional Generative Adversarial Network.

## 2. Анализ литературы

Архитектуру генеративной состязательной сети (generative adversarial network – GAN) предложил Ян Дж. Гудфеллоу в 2014 году [3]. Модель оказалась нестабильной в обучении, возникало несколько проблем, например, mode collapse – нейросеть воспроизводила один и тот же результат независимо от входных данных. Это связано с распространённой проблемой, возникающей при обучении нейронных сетей затухающими градиентами, такую проблему

решает модификация функции потерь, описанная в [5], а также была предложена модификация Unrolled GAN: [6]. С распространением свёрточных сетей появилась ещё одна модификация: DCGAN – Deep Convolutional GAN [7].

В конечном итоге архитектура GAN подходит для конкретных задач – обучить одну модель для генерации любого изображения или сигнала не получится, вероятность возникновения проблем велика, а их причины зачастую необъяснимы.

Вторая половина работы посвящена анализу временных рядов с помощью нейросетей. В этой области широко применяются рекуррентные модели. У таких архитектур есть недостаток – быстрое затухание градиентов [9]. Поэтому наиболее часто на практике применяется архитектура LSTM, имеющая блок долговременной памяти, который позволяет решить эту проблему [8].

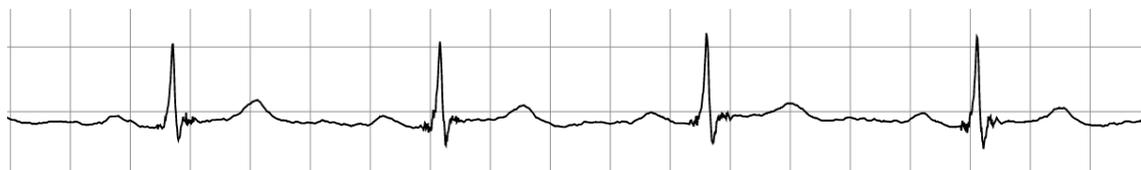
Задача классификации ударов сердца и устранения классового дисбаланса исследовалась в статье [2]. В работе проводится аналогичное исследование в учебных целях с некоторыми изменениями используемых архитектур, данных и другими результатами соответственно. В оригинальной статье используется аналогичное разбиение данных на обучающую и тестовую выборку, длина ЭКГ определяется как вектор из пространства столбцов длины 216. Аугментация проводилась для каждого класса по отдельности, работоспособность сети LSTM тестировалась на данных, которые были аугментированы для каждого класса. Результаты, полученные в оригинальной статье (точность предсказания каждого класса), представлены в таблице 1.

**Таблица 1.** Результаты статьи [2]

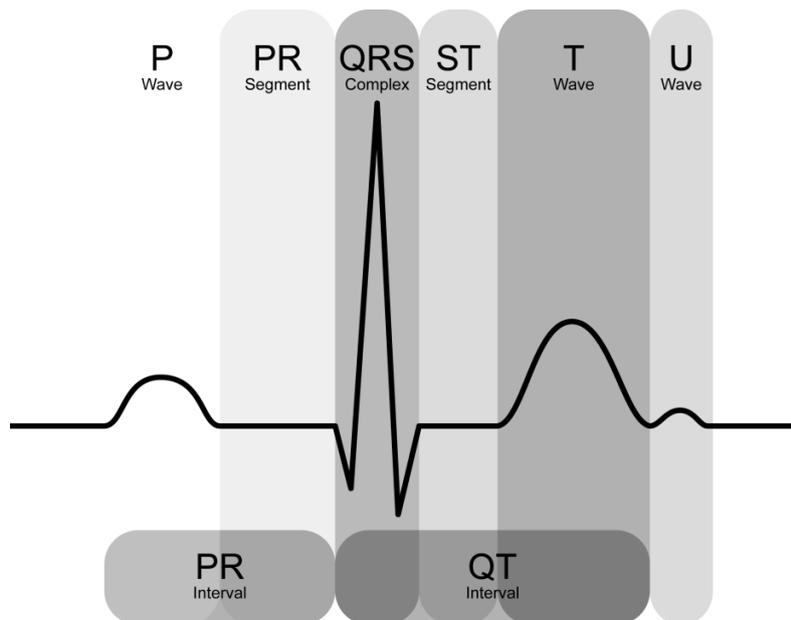
Класс	Без аугментации	С аугментацией DCGAN
0 / N	0.87 %	0.91 %
1 / S	0.82 %	0.92 %
2 / V	0.97 %	0.98 %
3 / F	0.95 %	0.96 %

### 3. Данные

Электрокардиограмма (ЭКГ) – запись электрической активности сердца. Снимается с помощью 12 датчиков с разных мест человеческого тела. Каждый сигнал состоит из ударов сердца, каждый из ударов состоит из *p*-волны, QRS комплекса, а также в некоторых случаях *и*-волны.

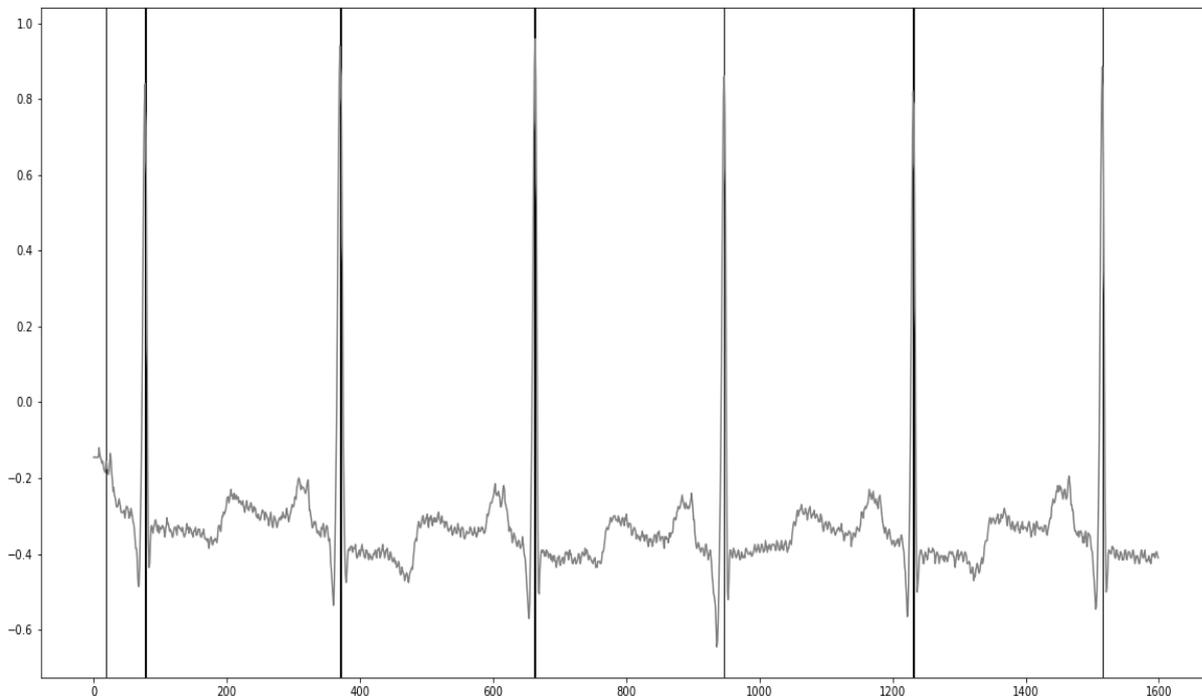


**Рис. 1.** Пример ЭКГ



**Рис. 2.** Пример сердечного сокращения

Для исследования использовалась база MIT-BIH Arrhythmia Database с записями ЭКГ [1]. В базе 30-минутные ЭКГ от 48 пациентов, есть информация о типе каждого сердечного удара, сами удары размечены на месте *R*-пиков. Каждая ЭКГ была сегментирована на удары сердца вручную, отступая от *R*-пика на равное расстояние влево и вправо. Также для дальнейшей работы в начало каждого удара добавлялось число – длительность в секундах.



**Рис. 3.** Пример данных из базы

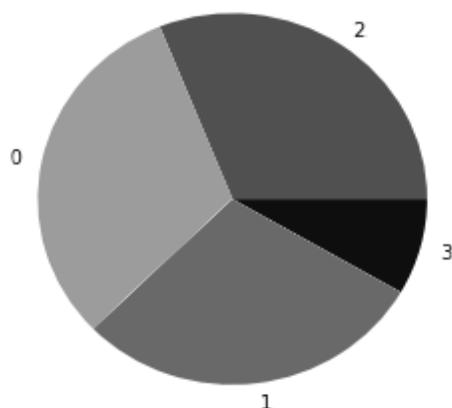
Для проведения классификации были взяты конкретные типы ударов и сформированы классы (см. таблицу 2).

**Таблица 2.** Распределение диагнозов по классам

Класс	Диагнозы
Нормальный удар N / 0	нормальный удар, блокада левой ножки пучка Гиса, блокада правой ножки пучка Гиса, предсердный выскальзывающий импульс, узловой ускользящий удар
Наджелудочковая экстрасистолия S / 1	предсердная экстрасистолия, аберрированная предсердная экстрасистолия, узловая экстрасистолия, наджелудочковый преждевременный удар
Желудочковая экстрасистолия V / 2	преждевременный желудочковый удар, выскальзывающий желудочковый импульс
Смешанный удар F / 3	слияние желудочкового и нормального удара

Разделение на обучающую и тестовую выборки происходило на уровне пациентов, то есть половина из них была отнесена к обучающей выборке, а вторая половина – к тестовой. Такой подход позволяет избежать утечки данных при обучении: идентичные ЭКГ будут находиться в разных выборках. После этого ЭКГ обрабатывались для получения ударов.

В оригинальном датасете сильно преобладает класс 0, в датасете для исследования равномерно отобрана только часть этих сигналов так, чтобы все диагнозы попали в тренировочную выборку. После такого преобразования стал виден дисбаланс между тремя остальными классами – смешанные удары в меньшинстве. Такая ситуация подходит для исследования влияния расширения обучающей выборки на точность предсказания каждого класса.



**Рис. 4.** Количественное распределение сигналов по классам

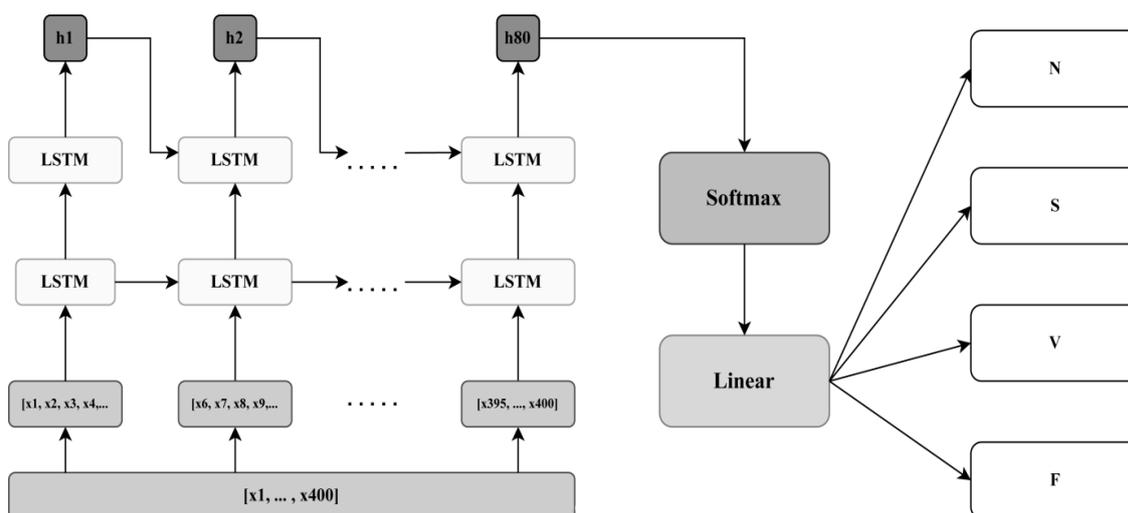


Рис. 5. Архитектура LSTM

## 4. Теоретическое обоснование

### 4.1 Архитектура LSTM

Сеть LSTM принимает на вход последовательность данных, таких как временные ряды или одномерный сигнал, каждый раз захватывая некоторый отрезок этой последовательности (длины 5 в рассматриваемой задаче). Каждый раз некоторая важная информация сохраняется в скрытом слое, который в данном случае представляет собой вектор длины 512.

Стоит отметить, что два ряда блоков LSTM – это одни и те же 2 блока в разные моменты времени. Это механика работы рекуррентной сети. Улучшение, которое предлагает рассмотренная архитектура, состоит в следующем: есть скрытый слой  $h$ , он же долговременная память. Его индексация означает разное состояние в разные моменты времени, сам слой единственен. Это позволяет решить проблему затухающих градиентов, которая возникает в обычной рекуррентной сети, поскольку она быстро "забывает" результаты предыдущих итераций, и метод обратного распространения ошибки не может зафиксировать каких-то различий между слоями.

### 4.2 Архитектура DCGAN

GAN или генеративно-сопоставительная сеть состоит из двух соревнующихся между собой моделей. Первая – генератор, из случайного вектора (шума) воссоздаёт изображение или сигнал и подаёт на вход другой сети – дискриминатору. Задача дискриминатора – это задача бинарной классификации: изображение реальное или искусственно созданное? Информация об этом поступает обратно генератору, за счёт чего обучается и он, и дискриминатор. Такой подход предшествовал современным генеративным нейросетям, которые в настоящее время совершают революцию в так называемом искусственном интеллекте. DCGAN – Deep Convolutional GAN использует свёрточные слои.

Как предложено в [3], генератор и дискриминатор играют в минимаксную игру

$$V(D, G) = E_{x \sim data} [\log \log D(x)] + E_{z \sim noise} [\log \log (1 - D(G(z)))].$$

В идеале нужно найти точку равновесия Нэша, от которой будет невыгодно отклоняться никому из двух соревнующихся сетей. В этой точке ошибка  $D$  будет равна 0.5: дискриминатор не может отличить реальные изображения от искусственных. Теоретически найти эту точку – сложная, до сих пор не изученная полностью задача. При обучении GAN часто возникают

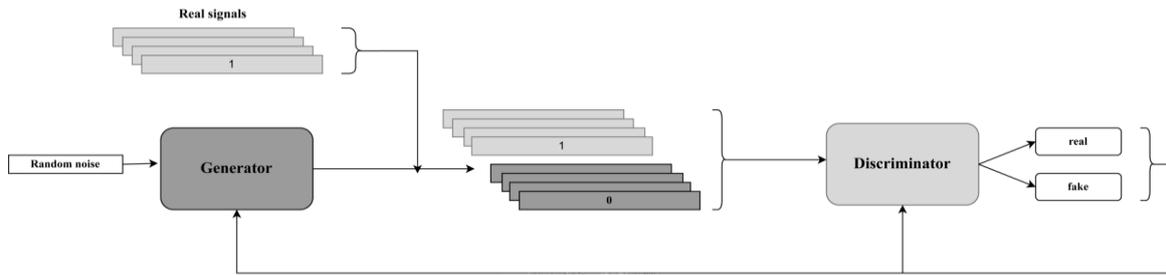


Рис. 6. Архитектура GAN

проблемы, не имеющие четко доказанного решения, которое всегда бы работало (например, mode collapse: генератор в попытке обмануть дискриминатор учится выдавать один и тот же паттерн выхода, который лучше всего "запутывает" этот самый дискриминатор).

С появлением диффузионных моделей, вероятно, нестабильность обучения GAN немного утратила свою актуальность.

При обучении отдельно тренируются генератор и дискриминатор. В цикле обучения генератора стоит задача максимизировать ошибку дискриминатора при искусственных входных данных:

$$\log \log D(G(z)).$$

После этого веса генератора замораживаются и начинается обучение дискриминатора  $D$ . Здесь уже стоит задача добиться максимальной точности классификации реальных и искусственных сигналов:

$$\log \log D(x) + \log \log (1 - D(G(z))).$$

Логарифм в этих выражениях объясняется тем, что в качестве функции потерь взята бинарная кросс-энтропия. Стоит отметить, что для разных ситуаций на каждой итерации генератор и дискриминатор можно тренировать разное количество раз.

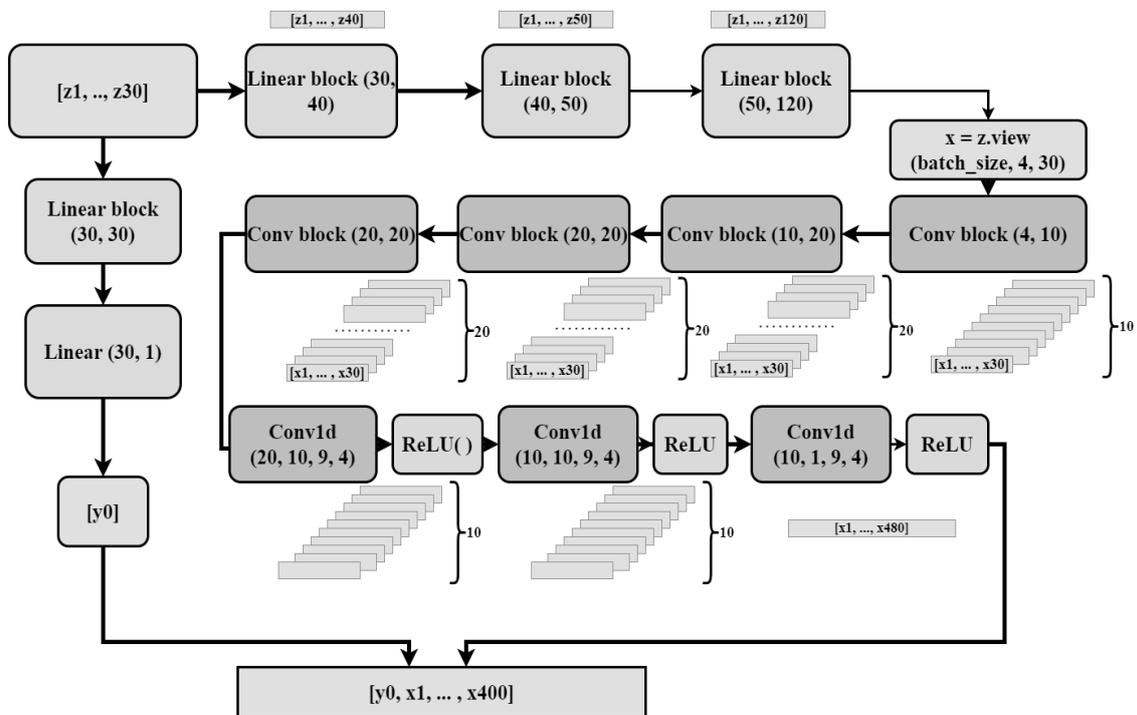


Рис 7. Архитектура генератора

В архитектуре генератора предлагается пропустить сигнал, представленный вектором длины 400, сначала через несколько полносвязных слоёв, затем через блоки с обратной свёрткой и нелинейностью, затем через обычные свёрточные слои и нелинейности.

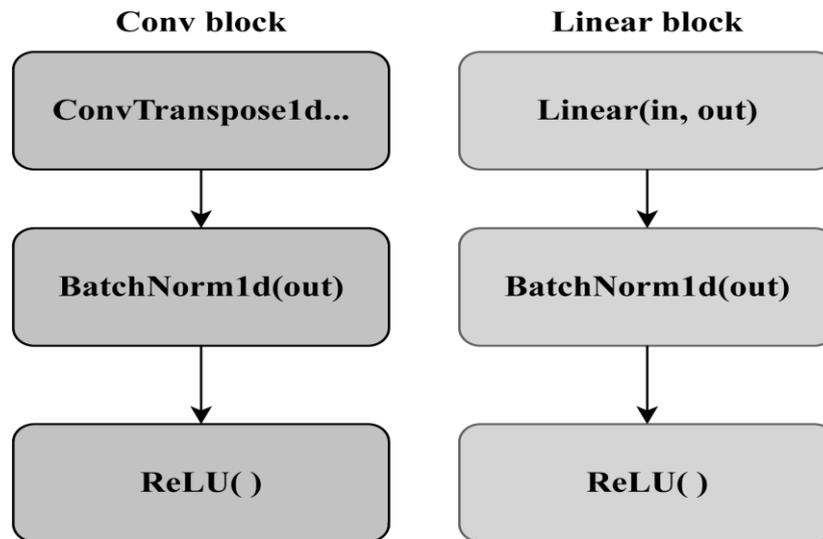


Рис. 8. Свёрточный и линейный блоки генератора

В отличие от предложенной в [2] архитектуры используется ещё длина ЭКГ в секундах, у каждого удара она своя, это помогло предотвратить *mode collapse* и получить уникальные сигналы для каждого случайного входного вектора. Входной шум используется дважды: для генерации ЭКГ и для генерации её длины.

Дискриминатор – это обычная свёрточная классифицирующая сеть, которая аналогично генератору отдельно работает с длиной ЭКГ, соединяя полученные результаты в конце и представляющей как выход для каждого объекта из батча вероятность принадлежности к классу 1 – класс реальных ЭКГ.

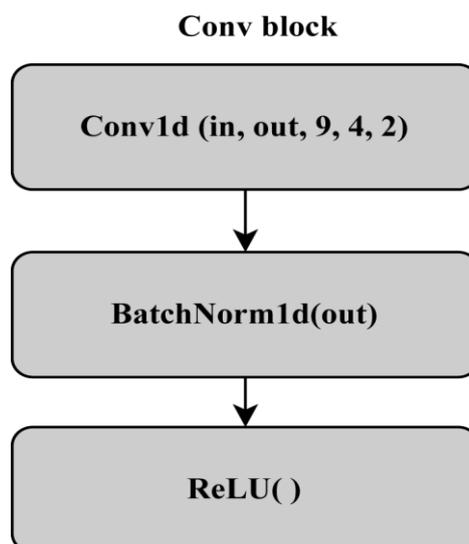


Рис. 9. Свёрточный блок дискриминатора

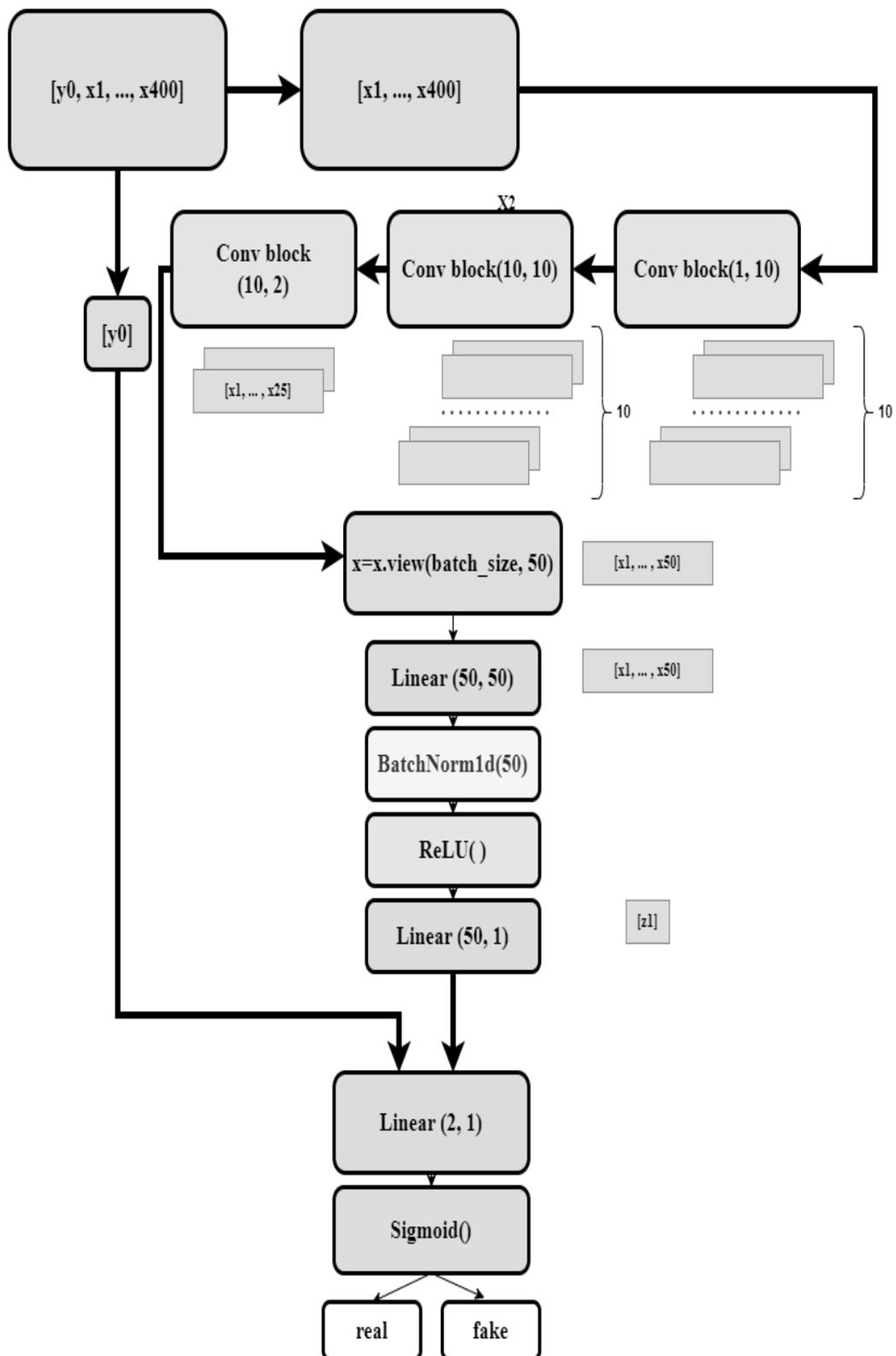


Рис. 10. Архитектура дискриминатора

## 5. Исследование

Параметры обучения сетей представлены в таблице 3.

Таблица 3. Параметры обучения сетей

LSTM		GAN (для аугментации класса 3)	
Размер батча	16	Размер батча	16
Число эпох	120	Число эпох	147
Функция потерь	MultiLabelSoftMarginLoss	Функция потерь	BinaryCrossEntropyLoss
Решающее правило(optimizer)	Adam	Решающее правило(optimizer)	Adam
Параметр t градиентного спуска	0.001	Параметр t градиентного спуска	0.0002
Beta1, beta2	0.9, 0.999	Beta1, beta2	0.5, 0.999
Размер скрытого слоя долговременной памяти	512	Размер входного случайного вектора	30
Размер участка сигнала	5	Устройство	NVIDIA GeForce GTX 1060 6GB
Устройство	NVIDIA GeForce GTX 1060 6GB		

Для аугментации разных классов выбиралось разное число эпох, а также разное количество циклов обучения генератора в расчёте на 1 итерацию.

## 6. Результаты исследования

В ходе исследования были аугментированы классы 1, 2 и 3, после чего добавлены в обучающую выборку в разных сочетаниях, далее фиксировались результаты обучения на неизменной тестовой выборке.

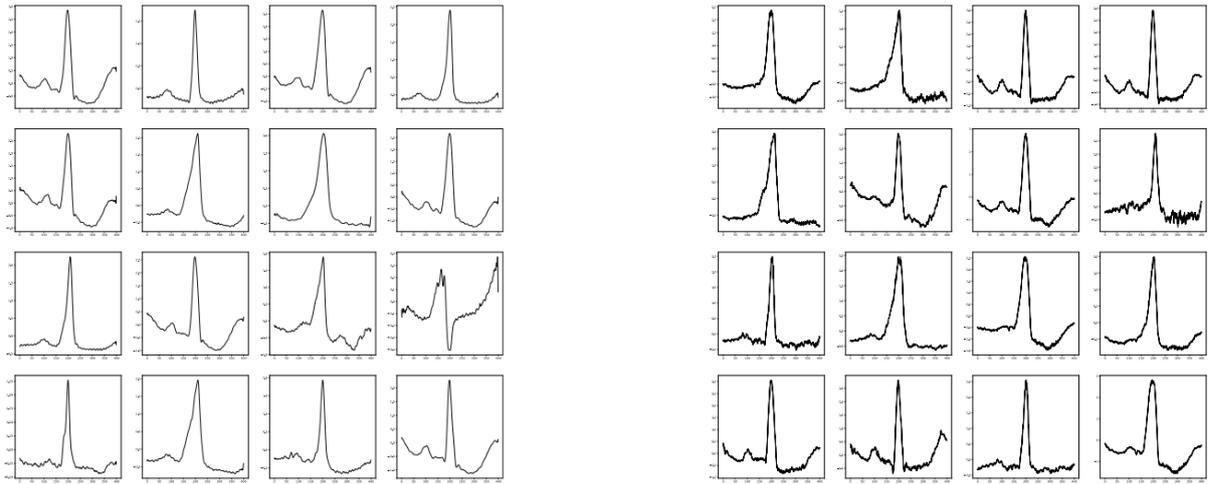


Рис. 11. Сравнение реальных ЭКГ (слева) и сгенерированных

Анализ кривой обучения GAN (рис. 12) позволяет увидеть, как эволюционировали генератор и дискриминатор. Как и ожидалось согласно теории, ошибка дискриминатора находится в положительной полу-окрестности значения 0.5, которое соответствует равновесию по Нэшу. Ошибка генератора больше, и наиболее реалистичные ЭКГ были получены в моменты максимальной близости двух графиков.

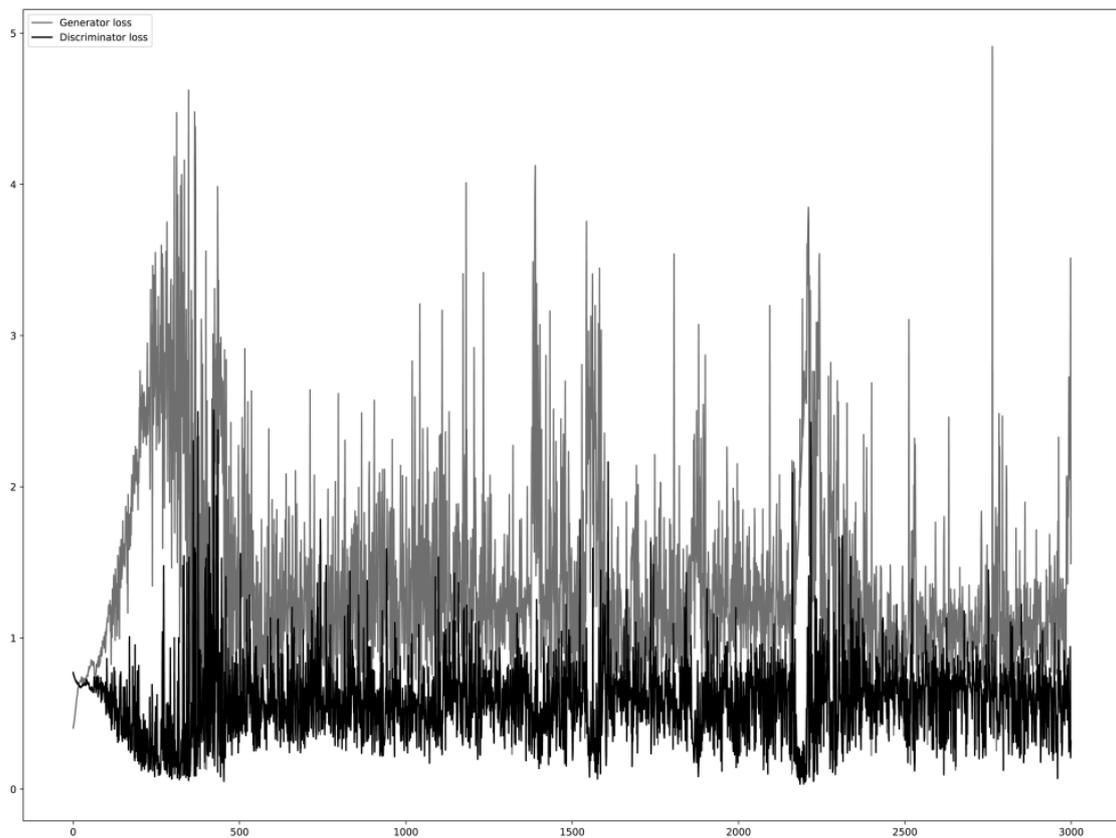


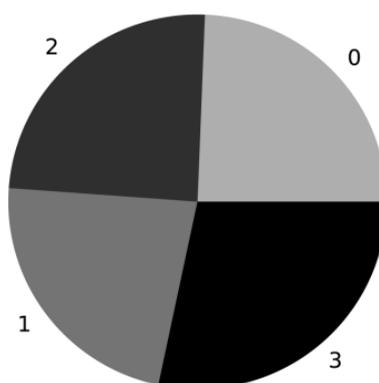
Рис. 12. Кривая обучения GAN, черный цвет - ошибка D, серый - ошибка G

Сеть LSTM обучалась 2 раза: до аугментации и после. Результаты – сравнение точности – представлены в таблице 4.

**Таблица 4.** Влияние аугментации на точность предсказания каждого класса

Класс	Без аугментации	Аугментация класса 3	Аугментация класса 1	Аугментация классов 1 и 3	Аугментация классов 1, 2 и 3
0 / N	75.9 %	71.2 %	62.9%	55.3 %	66.1 %
1 / S	15.7 %	17.3 %	<b>51.0%</b>	<b>39.7 %</b>	44.1 %
2 / S	85.2 %	83.9 %	87.0%	75.4 %	76.6 %
3 / F	0.5 %	<b>57.8 %</b>	0.8%	<b>30.9 %</b>	51.5 %

Распределение классов после аугментации отображено на рис. 13.



**Рис. 13.** Распределение классов после аугментации (классы 1, 2 и 3)

Изначально был несбалансирован только класс 3, однако не только у него оказалась крайне низкая точность предсказания (0.5%). Класс 1 также не даёт высоких результатов. В ходе эксперимента сначала была проведена аугментация класса 3, что подняло его точность предсказания до 57.8%. Аналогично была проведена аугментация только первого класса, что также подняло точность его предсказания до 51%. При попытке дополнить обучающую выборку сигналами из сразу 2 классов, точность предсказания обоих выросла, но результаты оказались хуже: для классов 1 и 3 точности 39.7% и 30.9% соответственно. Кроме того, при таком способе немного снизилась точность предсказания класса 2. При попытке аугментации классов 1, 2 и 3 точность предсказания всех классов опять выросла.

## 7. Заключение

Аугментация классов-патологий привела к значительному повышению точности предсказания каждого из них, однако в зависимости от набора классов, для которых проводилась аугментация, точности меняются. Однако, ввиду маленького числа ЭКГ в обучающей выборке (24 экземпляра), даже с аугментацией полученные точности не могут быть применимы для практического использования.

В связи с этой проблемой есть несколько планируемых улучшений. Во-первых, нужно собрать больше размеченных врачами данных для обучения генеративно-сопоставительной сети. Во-вторых, планируется улучшить архитектуру GAN с использованием слоёв Embedding, чтобы не обучать нейросеть заново для каждого класса, а иметь возможность генерации какой-то патологии по запросу (номеру класса). Предполагается, что с этими улучшениями искусственные ЭКГ будут лучше отражать свой класс, и в целом, будут более разнообразными, что в конечном итоге приведёт к повышению точности предсказания всех классов.

## Литература

1. Moody G. B., Mark R. G. The impact of the MIT-BIH arrhythmia database //IEEE engineering in medicine and biology magazine. – 2001. – Т. 20. – № 3. – С. 45-50.
2. Golany T. et al. Improving ECG classification using generative adversarial networks //Proceedings of the AAAI Conference on Artificial Intelligence. 2020. Т. 34. № 8. С. 13280-13285.
3. Goodfellow I. et al. Generative adversarial networks //Communications of the ACM. – 2020. – Т. 63. – № 11. – С. 139-144.
4. Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – № 8. – С. 1735-1780.
5. Arjovsky M., Chintala S., Bottou L. Wasserstein generative adversarial networks //International conference on machine learning. – PMLR, 2017. – С. 214-223.
6. Metz L. et al. Unrolled generative adversarial networks //arXiv preprint arXiv:1611.02163. – 2016.
7. Radford A., Metz L., Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks //arXiv preprint arXiv:1511.06434. – 2015.
8. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network //Physica D: Nonlinear Phenomena. – 2020. – Т. 404. – С. 132306.
9. Pascanu R., Mikolov T., Bengio Y. On the difficulty of training recurrent neural networks //International conference on machine learning. – Pmlr, 2013. – С. 1310-1318.

# Исследование классов групп, замкнутых относительно операций

В.Н. Сорокина, М.М. Сорокина

Брянский государственный университет имени академика И.Г. Петровского

Теория конечных групп является одним из основных разделов современной алгебры. В рамках данной теории в настоящее время интенсивно развивается ряд направлений, среди которых большую роль играет теория классов групп. Большое значение в теории классов групп занимают вопросы, связанные с исследованием операций на классах групп. Хорошо известными операциями на классах групп являются операции  $Q$  и  $R_0$ , определяющие центральные объекты в теории классов конечных групп, называемые формациями. Среди формаций наиболее изученными являются локальные формации, естественным обобщением которых являются  $\omega$ -локальные формации, где  $\omega$  – некоторое множество простых чисел.

*Ключевые слова:* группа, конечная группа, класс групп, операция на классах групп, нормальная подгруппа, факторгруппа.

## 1. Введение

Одной из важных задач алгебры является исследование операций над рассматриваемыми объектами: группами, кольцами, полями и т. д. «Алгебра может быть определена как наука о системах объектов той или иной природы, в которых установлены операции, по своим свойствам более или менее сходные со сложением и умножением чисел» [1].

Теория конечных групп является одним из основных разделов современной алгебры (см., напр., [6]). В рамках данной теории в настоящее время интенсивно развивается ряд направлений, среди которых большую роль играет теория классов групп. Фундаментальные результаты о классах групп принадлежат Л.А. Шеметкову, А.Н. Скибе, В.С. Монахову, В.А. Ведерникову, К. Дерку, Т. Хоуксу, В. Го, А. Баллестер-Болинше и др. (см., напр, [6-10]).

Большое значение в теории классов групп занимают вопросы, связанные с исследованием операций на классах групп [2]. Хорошо известными операциями на классах групп являются операции  $Q$  и  $R_0$ , определяющие центральные объекты в теории классов конечных групп, называемые формациями [9].

Среди формаций наиболее изученными являются локальные формации [4], естественным обобщением которых являются  $\omega$ -локальные формации, где  $\omega$  – некоторое множество простых чисел. При изучении и применении  $\omega$ -локальных формаций были введены в рассмотрение новые виды классов групп с учётом множества  $\omega$ , например,  $\omega$ -примитивно замкнутые классы групп и др. [13]. В настоящее время актуальной является задача рассмотрения аналогов операций  $Q$  и  $R_0$  с учётом множества  $\omega$ .

Научная новизна. Все полученные результаты являются новыми: установлены новые свойства операций  $Q^\omega$  и  $R_0^\omega$  на классах групп, новые свойства  $Q^\omega R_0^\omega$ -замкнутых классов групп. Результаты исследования опубликованы в [10, 11].

Методы исследования. В работе используются классические методы доказательств теории групп, а также методы теории классов конечных групп.

## 2. Основная часть

### Предварительные сведения

Рассматриваются только конечные группы. Используемые обозначения и определения для групп и классов групп стандартны (см. [7]). Приведем лишь некоторые из них. В дальнейшем

символ  $G$  обозначает произвольную конечную группу. Запись  $H \triangleleft G$  означает, что  $H$  – нормальная подгруппа группы  $G$ .

Пусть  $\mathbb{P}$  – множество всех простых чисел,  $\omega$  – непустое подмножество множества  $\mathbb{P}$ . Натуральное число  $n$  называется  $\omega$ -числом, если  $n$  делится только на простые числа из  $\omega$ . Через  $|G|$  обозначается порядок конечной группы  $G$ , т.е. количество элементов группы  $G$ ,  $\pi(G)$  – совокупность всех простых делителей порядка группы  $G$ . Группа  $G$  называется  $\omega$ -группой, если  $\pi(G) \subseteq \omega$ . Подгруппа  $H$  группы  $G$  называется  $\omega$ -подгруппой группы  $G$ , если  $H$  является  $\omega$ -группой.

Классом групп называется множество групп, содержащее вместе с каждой своей группой  $G$  и все группы, изоморфные  $G$ . Операцией на классах групп называется отображение множества всех классов групп в себя.

Пусть  $\Sigma$  – множество всех классов групп.

Операцией  $Q^\omega$  на классах групп называется отображение  $Q^\omega: \Sigma \rightarrow \Sigma$ , заданное по правилу:

$Q^\omega \mathfrak{X} = \{G \mid \exists H \in \mathfrak{X}, \exists N \triangleleft H, |N| - \omega\text{-число}, G \cong H/N\}$  для любого класса  $\mathfrak{X} \in \Sigma$ .

Операцией  $R_0^\omega$  на классах групп называется отображение  $R_0^\omega: \Sigma \rightarrow \Sigma$ , заданное по правилу:

$R_0^\omega \mathfrak{X} = \{G \mid G \cong H/(N_1 \cap \dots \cap N_t), \text{ где } H/N_i \in \mathfrak{X}, |N_i| - \omega\text{-число}, i = 1, \dots, t, t \geq 2\}$  для любого класса  $\mathfrak{X} \in \Sigma$  или, иначе,

$G \in R_0^\omega \mathfrak{X} \Leftrightarrow \exists N_i \triangleleft G, G/N_i \in \mathfrak{X}, |N_i| - \omega\text{-число}, i = \overline{1, t}, t \geq 2, N_1 \cap \dots \cap N_t = E$ , где  $E$  – единичная подгруппа.

Класс групп  $\mathfrak{F}$  называется:

$Q^\omega$ -замкнутым, если из того, что  $G \in \mathfrak{F}$  и  $N$  – нормальная  $\omega$ -подгруппа группы  $G$ , всегда следует, что  $G/N \in \mathfrak{F}$ ;

$R_0^\omega$ -замкнутым, если из того, что  $G/N_1, G/N_2 \in \mathfrak{F}$  и  $N_1, N_2$  – нормальные  $\omega$ -подгруппы группы  $G$ , всегда следует, что  $G/(N_1 \cap N_2) \in \mathfrak{F}$ .

Класс групп  $\mathfrak{F}$  называется  $\omega$ -формацией, если класс  $\mathfrak{F}$  является  $Q^\omega R_0^\omega$ -замкнутым.

Замечание 1. Всякая формация является  $\omega$ -формацией для любого непустого множества  $\omega$  простых чисел. Если  $\omega = \mathbb{P}$ , то понятие  $\omega$ -формации групп совпадает с понятием формации групп.

Пусть  $\mathfrak{F}$  – непустая формация групп,  $G$  – группа.  $\mathfrak{F}^\omega$ -корадикалом группы  $G$  называется пересечение всех нормальных  $\omega$ -подгрупп группы  $G$ , фактор-группы по которым принадлежат  $\mathfrak{F}$ , и обозначается  $G^{\mathfrak{F}^\omega}$ .

Замечание 2. Пусть  $\mathfrak{F}$  – непустая  $\omega$ -формация,  $G$  – группа. Тогда  $\mathfrak{F}^\omega$ -корадикал группы  $G$  является наименьшей нормальной  $\omega$ -подгруппой группы  $G$ , фактор-группа по которой принадлежит  $\mathfrak{F}$ , т.е. подгруппа  $G^{\mathfrak{F}^\omega}$  удовлетворяет следующим трем условиям:

- 1)  $G^{\mathfrak{F}^\omega} \triangleleft G, |G^{\mathfrak{F}^\omega}| - \omega\text{-число};$
- 2)  $G/G^{\mathfrak{F}^\omega} \in \mathfrak{F};$
- 3)  $G^{\mathfrak{F}^\omega}$  – наименьшая подгруппа группы  $G$ , удовлетворяющая условиям 1) и 2).

Замечание 3.  $\mathfrak{F}$ -Корадикал любой группы содержится в её  $\mathfrak{F}^\omega$ -корадикале, для любого непустого множества  $\omega$  простых чисел. Если  $\omega = \mathbb{P}$ , то понятие  $\mathfrak{F}^\omega$ -корадикала группы совпадает с понятием  $\mathfrak{F}$ -корадикала группы.

## 1. Исследование свойств операции $Q^\omega$ на классах групп

**Теорема 1.1.** Пусть  $\mathfrak{X}$  – класс групп. Тогда  $\mathfrak{X} \subseteq Q^\omega \mathfrak{X}$ .

*Доказательство.* Пусть  $G \in \mathfrak{X}$ . Покажем, что  $G \in Q^\omega \mathfrak{X}$ . Так как  $G \in \mathfrak{X}$ ,  $E \triangleleft G$ ,  $|E| = 1$ , то  $\pi(E) = \emptyset \subseteq \omega$  и поэтому  $|E| - \omega$ -число. Тогда  $G \cong G/E$  и с учётом определения операции  $Q^\omega$  справедливо  $G \in Q^\omega \mathfrak{X}$ . Таким образом,  $\mathfrak{X} \subseteq Q^\omega \mathfrak{X}$ .

Теорема доказана.

**Теорема 1.2.** Пусть  $\mathfrak{X}$  – класс групп. Тогда  $Q^\omega \mathfrak{X} = (Q^\omega)^2 \mathfrak{X}$ .

*Доказательство.* Покажем, что  $Q^\omega \mathfrak{X} = (Q^\omega)^2 \mathfrak{X}$ .

а) По теореме 1.1 имеем  $Q^\omega \mathfrak{X} = \mathfrak{X}_1 \subseteq Q^\omega \mathfrak{X}_1 = Q^\omega(Q^\omega \mathfrak{X}) = (Q^\omega)^2 \mathfrak{X}$ . Таким образом,  $Q^\omega \mathfrak{X} \subseteq (Q^\omega)^2 \mathfrak{X}$ .

б) Установим, что  $(Q^\omega)^2 \mathfrak{X} \subseteq Q^\omega \mathfrak{X}$ . Пусть  $G \in (Q^\omega)^2 \mathfrak{X}$ . Проверим, что  $G \in Q^\omega \mathfrak{X}$ .

Пусть  $Q^\omega \mathfrak{X} = \mathfrak{X}_1$ . Так как  $G \in (Q^\omega)^2 \mathfrak{X} = Q^\omega(Q^\omega \mathfrak{X})$ , то существует группа  $H \in \mathfrak{X}_1$  такая, что найдётся  $N \triangleleft H$ ,  $|N| - \omega$ -число и  $G \cong H/N$ .

Поскольку  $H \in \mathfrak{X}_1 = Q^\omega \mathfrak{X}$ , то существует группа  $K \in \mathfrak{X}$  такая, что найдётся  $L \triangleleft K$ ,  $|L| - \omega$ -число и  $H \cong K/L$ .

Так как  $H \cong K/L$  и  $N \triangleleft H$ , то существует  $(M/L) \triangleleft (K/L)$  такая, что  $N \cong M/L$ . Тогда  $H/N \cong (K/L)/(M/L) \cong K/M$ . Следовательно, по теореме о соответствии  $M \triangleleft K$ .

Покажем, что  $|M| - \omega$ -число. Действительно, поскольку  $M/L \cong N$  и  $|N| - \omega$ -число, то  $|M/L| - \omega$ -число. Так как  $|L| - \omega$ -число и  $|M/L| - \omega$ -число, то  $|M| - \omega$ -число.

Поскольку  $G \cong H/N$ , то  $G \cong K/M$ , где  $K \in \mathfrak{X}$ ,  $M \triangleleft K$ ,  $|M| - \omega$ -число. Следовательно,  $G \in Q^\omega \mathfrak{X}$ .

Таким образом,  $(Q^\omega)^2 \mathfrak{X} \subseteq Q^\omega \mathfrak{X}$ .

Из а) и б) следует равенство  $Q^\omega \mathfrak{X} = (Q^\omega)^2 \mathfrak{X}$ .

Теорема доказана.

В случае, когда  $\omega = \mathbb{P}$ , из теорем 1.1 и 1.2 вытекают следующие известные свойства операции  $Q$  на классах групп.

**Следствие 1.1** ([2], гл. 2, с. 22). Пусть  $\mathfrak{X}$  – класс групп. Тогда  $\mathfrak{X} \subseteq Q\mathfrak{X}$ .

**Следствие 1.2** ([2], гл. 2, с. 22). Пусть  $\mathfrak{X}$  – класс групп. Тогда  $Q\mathfrak{X} = Q^2\mathfrak{X}$ .

## 2. Исследование свойств операции $R_0^\omega$ на классах групп

**Теорема 2.1.** Пусть  $\mathfrak{X}$  – класс групп. Тогда  $\mathfrak{X} \subseteq R_0^\omega \mathfrak{X}$ .

*Доказательство.* Пусть  $G \in \mathfrak{X}$ . Покажем, что  $G \in R_0^\omega \mathfrak{X}$ . Пусть  $N_1 = E, N_2 = E, \dots, N_t = E$ . Тогда  $N_1, N_2, \dots, N_t$  – нормальные  $\omega$ -подгруппы группы  $G$ . Поскольку

$$G/N_1 \cong G \in \mathfrak{X}, \quad G/N_2 \cong G \in \mathfrak{X}, \quad \dots, \quad G/N_t \cong G \in \mathfrak{X},$$

то  $G \cong G/(N_1 \cap N_2 \dots N_t) \in R_0^\omega \mathfrak{X}$ . Таким образом,  $\mathfrak{X} \subseteq R_0^\omega \mathfrak{X}$ .

Теорема доказана.

**Теорема 2.2.** Пусть  $\mathfrak{X}$  – класс групп. Тогда  $R_0^\omega \mathfrak{X} = (R_0^\omega)^2 \mathfrak{X}$ .

*Доказательство.* Покажем, что  $R_0^\omega \mathfrak{X} = (R_0^\omega)^2 \mathfrak{X}$ .

а) Установим, что  $R_0^\omega \mathfrak{X} \subseteq (R_0^\omega)^2 \mathfrak{X}$ . Действительно, по теореме 2.1 справедливо

$$R_0^\omega \mathfrak{X} = \mathfrak{X}_1 \subseteq R_0^\omega \mathfrak{X}_1 = R_0^\omega(R_0^\omega \mathfrak{X}) = (R_0^\omega)^2 \mathfrak{X}.$$

Следовательно,  $R_0^\omega \mathfrak{X} \subseteq (R_0^\omega)^2 \mathfrak{X}$ .

б) Покажем, что  $(R_0^\omega)^2 \mathfrak{X} \subseteq R_0^\omega \mathfrak{X}$ . Пусть  $G \in (R_0^\omega)^2 \mathfrak{X} = R_0^\omega(R_0^\omega \mathfrak{X})$ . Тогда существует группа  $N_i \triangleleft G$  такая, что  $G/N_i \in R_0^\omega \mathfrak{X}$ ,  $|N_i| - \omega$ -число,  $i = \overline{1, t}$ ,  $N_1 \cap \dots \cap N_t = E$ .

Так как  $G/N_i \in R_0^\omega \mathfrak{X}$ , то

$$(a_1) \quad \exists L_{11}/N_1 \triangleleft G/N_1, \exists L_{12}/N_1 \triangleleft G/N_1, \dots, \exists L_{1r_1}/N_1 \triangleleft G/N_1,$$

$$(b_1) \quad (G/N_1)/(L_{1j_1}/N_1) \in \mathfrak{X}, \quad j_1 = 1, 2, \dots, r_1, \quad r_1 \geq 2,$$

(c<sub>1</sub>)  $|L_{1j_1}/N_1|$  –  $\omega$ -число,  $j_1 = 1, 2, \dots, r_1$ ,  $r_1 \geq 2$ ,

(d<sub>1</sub>)  $L_{11}/N_1 \cap L_{12}/N_1 \cap \dots \cap L_{1r_1}/N_1 = N_1/N_1$ ,

и так далее,

(a<sub>t</sub>)  $\exists L_{t1}/N_t \triangleleft G/N_t, \exists L_{t2}/N_t \triangleleft G/N_t, \dots, \exists L_{tr_t}/N_t \triangleleft G/N_t$ ,

(b<sub>t</sub>)  $(G/N_t)/(L_{tj_t}/N_t) \in \mathfrak{X}$ ,  $j_t = 1, 2, \dots, r_t$ ,  $r_t \geq 2$ ,

(c<sub>t</sub>)  $|L_{1j_t}/N_t|$  –  $\omega$ -число,  $j_t = 1, 2, \dots, r_t$ ,

(d<sub>t</sub>)  $L_{t1}/N_t \cap \dots \cap L_{tr_t}/N_t = N_t/N_t$ .

Из (a<sub>1</sub>) – (a<sub>t</sub>) по теореме о соответствии следует, что

(a)  $L_{11}, \dots, L_{1r_1}, \dots, L_{t1}, \dots, L_{tr_t}$  – нормальные подгруппы группы  $G$ .

Из (b<sub>1</sub>) – (b<sub>t</sub>) по теореме о гомоморфизмах следует, что

(b)  $G/L_{11} \in \mathfrak{X}, \dots, G/L_{tr_t} \in \mathfrak{X}$ .

Из (c<sub>1</sub>) – (c<sub>t</sub>) по теореме Лагранжа

(c)  $|L_{1j_1}|$  –  $\omega$ -число,  $j_1 = 1, 2, \dots, r_1$ ,  $|L_{1j_t}|$  –  $\omega$ -число,  $j_t = 1, 2, \dots, r_t$ .

Из (d<sub>1</sub>) – (d<sub>t</sub>) следует, что  $L_{11} \cap \dots \cap L_{1r_1} = N_1, \dots, L_{t1} \cap \dots \cap L_{tr_t} = N_t$  и поэтому

(d)  $L_{11} \cap \dots \cap L_{tr_t} = N_1 \cap \dots \cap N_t = E$ .

Из (a) – (d) по определению операции  $R_0^\omega$  следует, что  $G \in R_0^\omega \mathfrak{X}$ , откуда  $(R_0^\omega)^2 \mathfrak{X} \subseteq R_0^\omega \mathfrak{X}$ .

Из а) и б) получаем равенство  $(R_0^\omega)^2 \mathfrak{X} = R_0^\omega \mathfrak{X}$ .

Теорема доказана.

В случае, когда  $\omega = \mathbb{P}$ , из теорем 2.1 и 2.2 вытекают следующие известные свойства операции  $R_0$  на классах групп.

**Следствие 2.1** ([2], гл. 2, с. 22). Пусть  $\mathfrak{X}$  – класс групп. Тогда  $\mathfrak{X} \subseteq R_0 \mathfrak{X}$ .

**Следствие 2.2** ([2], гл. 2, с. 22). Пусть  $\mathfrak{X}$  – класс групп. Тогда  $Q\mathfrak{X} = R_0^2 \mathfrak{X}$ .

### 3. Исследование свойств $\omega$ -формаций

**Теорема 3.1.** Пересечение любой совокупности  $\omega$ -формаций является  $\omega$ -формацией.

*Доказательство.* Пусть  $\{\mathfrak{F}_i \mid i \in I\}$  – некоторая совокупность  $\omega$ -формаций и  $\mathfrak{F} = \bigcap_{i \in I} \mathfrak{F}_i$ . Покажем, что  $\mathfrak{F}^\omega$  является  $\omega$ -формацией. Если  $\mathfrak{F}_i = \emptyset$  для некоторого  $j \in I$ , то  $\mathfrak{F} = \emptyset$  –  $\omega$ -формация. Пусть  $\mathfrak{F}_i \neq \emptyset$  для любого  $i \in I$ . Проверим, что класс  $\mathfrak{F}$  удовлетворяет определению  $\omega$ -формации.

Пусть  $G \in \mathfrak{F}$  и  $N$  – нормальная  $\omega$ -подгруппа группы  $G$ . Тогда  $G \in \mathfrak{F}_i$  для любого  $i \in I$ . Поскольку класс  $\mathfrak{F}_i$  является  $\omega$ -формацией, то  $G/N \in \mathfrak{F}_i$  для любого  $i \in I$  и поэтому  $G/N \in \bigcap_{i \in I} \mathfrak{F}_i$ . Таким образом,  $G/N \in \mathfrak{F}$ .

Пусть  $G/N_1 \in \mathfrak{F}$ ,  $G/N_2 \in \mathfrak{F}$ , где  $N_1$  и  $N_2$  – нормальные  $\omega$ -подгруппы группы  $G$ . Тогда  $G/N_1 \in \mathfrak{F}_i$  и  $G/N_2 \in \mathfrak{F}_i$ , для любого  $i \in I$ . Поскольку класс  $\mathfrak{F}_i$  является  $\omega$ -формацией, то  $G/(N_1 \cap N_2) \in \mathfrak{F}_i$  для любого  $i \in I$  и поэтому  $G/(N_1 \cap N_2) \in \bigcap_{i \in I} \mathfrak{F}_i$ . Таким образом,  $G/(N_1 \cap N_2) \in \mathfrak{F}$ .

Тем самым установлено, что класс  $\mathfrak{F}$  является  $\omega$ -формацией.

Теорема доказана.

**Теорема 3.2.** Пусть  $\mathfrak{F}$  – непустая  $\omega$ -формация,  $G$  – группа. Тогда  $G \in \mathfrak{F}$  в том и только том случае, когда  $G^{\mathfrak{F}^\omega} = E$ .

*Доказательство.* I. *Необходимость.* Пусть  $G \in \mathfrak{F}$ . Покажем, что  $G^{\mathfrak{F}^\omega} = E$ . Так как  $G \cong G/E$ ,  $G \in \mathfrak{F}$  и  $\mathfrak{F}$  – класс групп, то по определению класса групп  $G/E \in \mathfrak{F}$ . Поскольку  $E$  – нормальная  $\omega$ -подгруппа группы  $G$ ,  $G/E \in \mathfrak{F}$  и  $E$  – наименьшая подгруппа группы  $G$ , то по замечанию 2  $G^{\mathfrak{F}^\omega} = E$ .

II. *Достаточность.* Пусть  $G^{\mathfrak{F}^\omega} = E$ . Так как  $G/G^{\mathfrak{F}^\omega} \in \mathfrak{F}$ , то  $G/E \in \mathfrak{F}$ . Поскольку  $G/E \cong G$ , то по определению класса групп  $G \in \mathfrak{F}$ .

Теорема доказана.

**Теорема 3.3.** Пусть  $G$  – группа,  $\mathfrak{F}$  –  $\omega$ -формація,  $K$  – нормальная  $\omega$ -подгруппа группы  $G$ . Тогда  $(G/K)^{\mathfrak{F}^\omega} = G^{\mathfrak{F}^\omega} K/K$ .

*Доказательство.* 1. Докажем равенство  $(G/K)^{\mathfrak{F}^\omega} = G^{\mathfrak{F}^\omega} K/K$  методом встречных включений. Обозначим  $(G/K)^{\mathfrak{F}^\omega}$  через  $N/K$ , т.е. полагаем, что

$$(G/K)^{\mathfrak{F}^\omega} = N/K. \quad (1)$$

1) Покажем, что  $G^{\mathfrak{F}^\omega} K/K \subseteq N/K$ . Для этого предварительно установим, что  $G^{\mathfrak{F}^\omega} \subseteq N$ . Ввиду замечания 2, достаточно проверить, что а)  $N$  – нормальная  $\omega$ -подгруппа группы  $G$ ; б)  $G/N \in \mathfrak{F}$ .

а) Докажем, что  $N$  – нормальная  $\omega$ -подгруппа группы  $G$ . По замечанию 2 (1)  $(G/K)^{\mathfrak{F}^\omega}$  – нормальная подгруппа группы  $G/K$ , то есть  $N/K$  – нормальная подгруппа в  $G/K$ . Отсюда по теореме о соответствии получаем, что  $N$  – нормальная подгруппа группы  $G$ .

Покажем, что  $|N|$  –  $\omega$ -число. Действительно, так как по замечанию 2  $|N/K|$  –  $\omega$ -число и по условию  $|K|$  –  $\omega$ -число, то по теореме Лагранжа  $|N|$  также является  $\omega$ -числом.

Таким образом,  $N$  является нормальной  $\omega$ -подгруппой группы  $G$ .

б) Установим, что  $G/N \in \mathfrak{F}$ . По теореме о гомоморфизмах

$$G/N \cong (G/K)/(N/K). \quad (2)$$

Ввиду (1), по замечанию 2 (2) справедливо

$$(G/K)/(N/K) \in \mathfrak{F}. \quad (3)$$

Так как  $\mathfrak{F}$  – класс групп, то из (2) и (3) следует, что  $G/N \in \mathfrak{F}$ .

Из а) и б) по замечанию 2 получаем, что  $G^{\mathfrak{F}^\omega} \subseteq N$ . Отсюда по теореме о соответствии следует, что  $G^{\mathfrak{F}^\omega} K/K \subseteq N/K$ .

2) Докажем, что  $N/K \subseteq G^{\mathfrak{F}^\omega} K/K$ . Исходя из (1), ввиду замечания 2, достаточно проверить, что а)  $G^{\mathfrak{F}^\omega} K/K$  – нормальная  $\omega$ -подгруппа группы  $G/K$ ; б)  $(G/K)/(G^{\mathfrak{F}^\omega} K/K) \in \mathfrak{F}$ .

а) Установим, что  $G^{\mathfrak{F}^\omega} K/K$  – нормальная  $\omega$ -подгруппа группы  $G/K$ . Действительно, по замечанию 2 (1) подгруппа  $G^{\mathfrak{F}^\omega}$  является нормальной  $\omega$ -подгруппой группы  $G$ . Отсюда по теореме о соответствии следует, что  $G^{\mathfrak{F}^\omega} K/K$  является нормальной в  $G/K$ .

Покажем, что  $|G^{\mathfrak{F}^\omega} K/K|$  –  $\omega$ -число. Действительно, так как по замечанию 2  $|G^{\mathfrak{F}^\omega}|$  –  $\omega$ -число и по теореме Лагранжа

$$|G^{\mathfrak{F}^\omega} K/K| = \frac{|G^{\mathfrak{F}^\omega} K|}{|K|} = \frac{|G^{\mathfrak{F}^\omega}| \cdot |K|}{|K| \cdot |K \cap G^{\mathfrak{F}^\omega}|} = \frac{|G^{\mathfrak{F}^\omega}|}{|K \cap G^{\mathfrak{F}^\omega}|},$$

то  $|G^{\mathfrak{F}^\omega} K/K|$  также является  $\omega$ -числом.

б) Покажем, что  $(G/K)/(G^{\mathfrak{F}^\omega} K/K) \in \mathfrak{F}$ . По теореме о гомоморфизмах имеют место следующие изоморфизмы

$$(G/K)/(G^{\mathfrak{F}^\omega} K/K) \cong G/G^{\mathfrak{F}^\omega} K, \quad (4)$$

$$G/G^{\mathfrak{F}^\omega} K \cong (G/G^{\mathfrak{F}^\omega})/(G^{\mathfrak{F}^\omega} K/G^{\mathfrak{F}^\omega}). \quad (5)$$

Согласно замечанию 2 (2),  $G/G^{\mathfrak{F}^\omega} \in \mathfrak{F}$ . Так как  $\mathfrak{F}$  – формація, то из по определению  $\omega$ -формації следует, что

$$(G/G^{\mathfrak{F}^\omega})/(G^{\mathfrak{F}^\omega} K/G^{\mathfrak{F}^\omega}) \in \mathfrak{F}. \quad (6)$$

Так как  $\mathfrak{F}$  – класс групп, то из (4), (5) и (6) по определению класса групп получаем  $(G/K)/(G^{\mathfrak{F}^\omega} K/K) \in \mathfrak{F}$ .

Из а) и б) по замечанию 2 заключаем, что  $N/K \subseteq G^{\mathfrak{F}^\omega} K/K$ .

Из 1) и 2) следует равенство  $N/K = G^{\mathfrak{F}^\omega} K/K$ . Таким образом, с учетом (1), приходим к выводу, что  $(G/K)^{\mathfrak{F}^\omega} = G^{\mathfrak{F}^\omega} K/K$ .

Теорема доказана.

В случае, когда  $\omega = \mathbb{P}$ , из теорем 3.1 – 3.3 вытекают следующие известные свойства формаций групп.

**Следствие 3.1** ([9], с. 9). *Пересечение любой совокупности формаций групп является формацией групп.*

**Следствие 3.2** ([7], с. 166). *Пусть  $\mathfrak{F}$  – непустая формация,  $G$  – группа. Тогда  $G \in \mathfrak{F}$  в том и только том случае, когда  $G^{\mathfrak{F}} = E$ .*

**Следствие 3.3** ([9], с. 9). *Пусть  $G$  – группа,  $\mathfrak{F}$  – формация,  $K$  – нормальная подгруппа группы  $G$ . Тогда  $(G/K)^{\mathfrak{F}} = G^{\mathfrak{F}} K/K$ .*

## Заключение

**Выводы.** В работе проведено исследование свойств классов групп, замкнутых относительно операций  $Q^\omega$  и  $R_0^\omega$ . При этом были решены следующие задачи:

- 1) установлены свойства операции  $Q^\omega$  на классах групп (теоремы 1.1 и 1.2);
- 2) установлены свойства операции  $R_0^\omega$  на классах групп (теоремы 2.1 и 2.2);
- 3) установлены свойства  $\omega$ -формаций (теоремы 3.1 – 3.3).

В случае, когда множество  $\omega$  совпадает с множеством  $\mathbb{P}$  всех простых чисел, из полученных теорем в качестве следствий вытекают известные свойства операций  $Q$  и  $R_0$  на классах групп, свойства формаций групп (см. [2, 7, 9]).

**Научно-практическая значимость результатов исследования.** Полученные результаты могут быть использованы при чтении спецкурсов, преподаваемых в высших учебных заведениях для студентов математических специальностей и направлений, а также могут быть использованы в дальнейших исследованиях по теории классов конечных групп студентами, магистрантами и аспирантами, специализирующимися в области алгебры.

## Литература

1. Большая Советская Энциклопедия. В 30-ти т., 3-е изд. – М.: Совет. энцикл., 1969 – 1986.
2. Ведерников В.А. Элементы теории классов групп. Смоленск: СГПУ, 1988.
3. Ведерников В.А., Сорокина М.М.  $\mathfrak{F}^\omega$ -нормализаторы конечных групп // Сиб. матем. журн., 2017. Т. 58, № 1. С. 64–82.
4. Воробьев Н.Н. Алгебра классов конечных групп. Витебск: ВГУ им. П.М. Машерова, 2012.
5. Каморников С.Ф., Селькин М.В. Подгрупповые функторы и классы конечных групп. – Минск: Беларуская навука, 2003.
6. Курош А.Г. Теория групп. – М.: Наука, 1967.
7. Монахов В.С. Введение в теорию конечных групп и их классов. Минск: Высшая школа, 2006.
8. Скиба А.Н. Алгебра формаций. Минск: Беларуская навука, 1997.
9. Шеметков Л.А. Формации конечных групп. М.: Наука, 1978.
10. Сорокина В.Н. О свойствах операции  $Q^\omega$  на классах групп // Актуальные проблемы современной науки: взгляд молодых учёных: материалы Национальной научно-практической студенческой конф., Т. 2, Брянск, 7-8 декабря 2022 г. / отв. ред. Е.Д. Селифонова, Т.А. Степченко, О.В. Тишина. – Брянск: РИСО БГУ, 2022. – С. 134–138.

11. Сорокина В.Н. О свойствах операции  $R_0^\omega$  на классах групп // Молодой ученый, 2022. № 51 (446). С. 3–6.
12. Doerk K., Hawkes T. Finite soluble groups. – Berlin – New York: Walter de Gruyter, 1992.
13. Guo W. The Theory of Classes of Groups. – Beijing – New York: Science Press, 2000.

# Об исследовании миноров матрицы инцидентности $d$ -мерного куба

*Е.Б. Титова*

*Нижегородский государственный университет им. Н.И. Лобачевского*

Для матрицы  $T$ , где  $T$  – матрица инцидентности вершины-фасеты  $d$ -мерного куба, получена асимптотика среднего значения суммы квадратов миноров рангового порядка  $j$  и в случае, когда порядок миноров  $j = \alpha d$ ,  $0 < \alpha < d$ . Проведено сравнение с аналогичной асимптотикой базисной системы строк матрицы  $T$ .

*Ключевые слова:* матрица инцидентности, асимптотика среднего значения минора, сингулярный многочлен.

## 1. Введение

Пусть  $A$  – целочисленная  $(m \times n)$ -матрица ( $m \leq n$ ),  $r$  – ее ранг,  $A^T$  – матрица, транспонированная к  $A$ .

Обозначим сумму квадратов миноров  $i$ -го порядка через  $s_i(A) = \sum \det^2 A(I, J)$ , где суммирование идёт по всем  $i$ -элементным подмножествам  $I \subseteq \{1, \dots, m\}$  и  $J \subseteq \{1, \dots, n\}$ , а  $A(I, J)$  – подматрица матрицы  $A$ , номера строк которой принадлежат подмножеству  $I \subseteq \{1, \dots, m\}$ , а номера столбцов – подмножеству  $J \subseteq \{1, \dots, n\}$ . Тогда характеристический многочлен матрицы  $AA^T$

$$\det(\lambda E_m - AA^T) = \sum_{i=0}^m (-1)^i s_i(A) \lambda^{m-i} = \lambda^{m-r} \sum_{i=0}^r (-1)^i s_i(A) \lambda^{r-i},$$

где  $\sum_{i=0}^r (-1)^i s_i(A) \lambda^{r-i} = \Delta(\lambda, A)$  – сингулярный многочлен матрицы  $A$ , а его корни – сингулярные числа.

Пусть выпуклый многогранник  $P$  описывается системой  $Ax \leq b, x \geq 0$ , где  $b$  – целочисленный вектор. Известно, что  $i$ -я координата вершины  $p$  многогранника  $P$  имеет в знаменателе число, равное по абсолютной величине минору  $j$ -го порядка матрицы  $A$ . Поэтому оценки величин миноров матрицы  $A$  представляют несомненный интерес в линейном, и особенно, в целочисленном линейном программировании.

## 2. Постановка задачи

Рассмотрим матрицу ограничений  $d$ -индексной аксиальной транспортной задачи при  $n = 2$  (обозначения и определение см., например, [1]). Матрицу  $T = T(d-1, d, 2)$  можно рассматривать по строчечным блокам  $T_i$  ( $i = 1, \dots, d$ ), тогда она имеет следующий вид:

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & \dots & 1 & 0 & \dots & 1 & 0 & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & 0 & 1 & \dots & 0 & 1 & \dots & 0 & 1 & 0 & 1 & \dots & 0 & 1 \\ 1 & 1 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 1 & 1 & \dots & 0 & 0 & 1 & 1 & \dots & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 1 & 1 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}.$$

Матрица  $T$  имеет ранг  $r = d + 1$ , число столбцов  $N = 2^d$ , число строк  $M = 2d$ .

Нетрудно видеть, что в этом случае матрицу  $T$  можно интерпретировать как матрицу инцидентности  $d$ -мерного куба,  $2^d$  столбцов которой задают все вершины куба,  $2d$  строк – грани

максимальной размерности (фасеты), 1 ставим в том и только том случае, когда вершина инцидентна фасете, и 0 в противном случае.

Известен ([2]) сингулярный многочлен матрицы  $T$ :

$$\Delta(\lambda, T) = (\lambda - 2^{d-1})^d \cdot (\lambda - d2^{d-1}). \quad (1)$$

Обозначим сумму квадратов миноров порядка  $j$  матрицы  $T$  через  $\sigma_j(d, T)$ , тогда среднее значение квадрата минора  $\delta_j(d, T)$  будет вычисляться по формуле

$$\delta_j(d, T_d) = \frac{\sigma_j(d, T)}{\binom{N}{j} \binom{M}{j}}. \quad (2)$$

Ранее автором рассматривалась асимптотика  $\delta_j(s, d, n)$  матрицы ограничений  $T(s, d, n)$  многоиндексной транспортной задачи при фиксированном  $d$ ,  $1 \leq s \leq d$  и  $n \rightarrow \infty$  для различных порядков  $j$  (см., напр., [1], [3]).

### 3. Основные результаты

Зафиксируем  $n = 2$ ,  $s = d - 1$  и рассмотрим поведение величины  $\delta_j(d, T)$  при  $d \rightarrow \infty$ .

**Теорема 1.** *Если  $d \rightarrow \infty$ , то*

$$\delta_r(d, T) \sim \frac{\sqrt{2} \pi d}{4} \left( \frac{d+1}{8e} \right)^{d+1}. \quad (3)$$

*Доказательство.* Будем использовать формулу Стирлинга в виде  $n! \sim \sqrt{2\pi n} \left( \frac{n}{e} \right)^n$ . Из (1) следует, что сумма квадратов миноров рангового порядка

$$\sigma_r(d, T) = d \cdot 2^{d^2-1}. \quad (4)$$

Вычислим количество миноров рангового порядка матрицы  $T_d$ .

$$\binom{N}{r} = \binom{2^d}{d+1} \sim \frac{2^{d(d+1)}}{(d+1)!} \sim \frac{2^{d^2+d}}{\sqrt{2\pi(d+1)}} \left( \frac{e}{d+1} \right)^{d+1}. \quad (5)$$

$$\binom{M}{r} = \binom{2d}{d+1} = \frac{d}{d+1} \binom{2d}{d} \sim \frac{1}{\sqrt{\pi d}} \cdot 4^d. \quad (6)$$

Подставляя выражения (4), (5), (6) в формулу (2), получим

$$\delta_r(d, T) \sim \frac{d \cdot 2^{d^2-1}}{2^{d^2+d} \cdot 4^d} \cdot \sqrt{2} \pi d \left( \frac{d+1}{e} \right)^{d+1} \sim \frac{\sqrt{2} \pi d}{4} \left( \frac{d+1}{8e} \right)^{d+1}. \quad (7)$$

Теорема доказана.

Формула (3) дает асимптотику величины  $\delta_r(d, T)$  при  $d \rightarrow \infty$ .

**Следствие 1.** *Среднее значение квадрата минора рангового порядка  $\delta_r(d, T)$  стремится к бесконечности при  $d \rightarrow \infty$  с экспоненциальной скоростью.*

**Следствие 2.** *Среднее значение минора рангового порядка матрицы  $T$  с ростом  $d$  стремится к бесконечности с экспоненциальной скоростью.*

*Доказательство.* Положим  $\varphi = \max |M_r|$ , где  $M_r$  – минор рангового порядка матрицы  $T$ . Тогда  $\sigma_r \leq \varphi \sigma_{1,r}$  (см., например, [4]), где  $\sigma_{1,r}$  – сумма модулей миноров рангового порядка матрицы  $T$ , а в рассматриваемом случае  $\varphi \leq 2^{-d} (d+1)^{\frac{d+1}{2}}$  (см. [5]). Поэтому

$$\frac{\sigma_{1,r}}{\binom{N}{r}\binom{M}{r}} \geq \left(\frac{d+1}{16e^2}\right)^{\frac{d+1}{2}}.$$

Отсюда следует утверждение следствия. Следствие доказано.

Рассмотрим случай  $j < r$ .

**Следствие 3.** При  $j = d - c$ , где постоянная  $c > 0$ , величина  $\delta_j(d, T) \rightarrow \infty$  с ростом  $d$ .

**Теорема 2.** Если  $d \rightarrow \infty$  и порядок миноров  $j = j(d)$ , то

$$\delta_j(d, T) \sim \sqrt{2\pi j} \cdot \left(\frac{j}{4e}\right)^j. \quad (8)$$

*Доказательство.* Из (1) следует, что

$$\sigma_j(d, T) = 2^{(d-1)j} \left[ \binom{d}{j} + \binom{d}{j-1} \right] = \frac{d+1}{d+1-j} \cdot 2^{(d-1)j} \binom{d}{j}. \quad (9)$$

Вычислим количество миноров  $j$ -го порядка матрицы  $T$ .

$$\binom{N}{j} = \binom{2^d}{j} = \frac{(2^d)!}{j!(2^d-j)!} = \frac{2^d \cdot \dots \cdot (2^d - j + 1)}{j!} \sim \frac{2^{dj}}{j!} \sim \frac{2^{dj}}{\sqrt{2\pi j}} \left(\frac{e}{j}\right)^j. \quad (10)$$

Тогда, подставляя выражения (9) и (10) в формулу (2), получим

$$\delta_j(d, T) \sim \sqrt{2\pi j} \frac{d+1}{d+1-j} \cdot \frac{2^{(d-1)j}}{2^{dj}} \cdot \left(\frac{j}{e}\right)^j \cdot \frac{\binom{d}{j}}{\binom{2^d}{j}} \sim \sqrt{2\pi j} \cdot \left(\frac{j}{2e}\right)^j \cdot \frac{1}{2^j} = \sqrt{2\pi j} \cdot \left(\frac{j}{4e}\right)^j.$$

Теорема доказана.

Формула (8) дает асимптотику величины  $\delta_j(d, T)$  при  $d \rightarrow \infty$ .

**Следствие 4.** При  $j = \alpha d$ ,  $0 < \alpha < 1$ , величина  $\delta_j(d, T) \rightarrow \infty$  при  $\alpha > \frac{4e}{d}$ , начиная с  $d \geq 11$ .

## 4. Заключение

В данной работе получена асимптотика среднего значения квадрата минора рангового и меньших порядков по всей матрице инцидентности вершины-фасеты  $d$ -мерного куба  $T$ . В [6] был получен сингулярный многочлен базисной системы строк матрицы  $T$ , что позволило в [7] оценить рост миноров рангового и меньших порядков этой системы строк. Сравнивая полученные результаты, видим, что в базисной системе строк рост миноров происходит быстрее.

## Литература

1. Титова Е.Б., Шевченко В.Н. Среднее значение квадрата минора матрицы ограничений аксиальной транспортной задачи. //Автоматика и телемеханика. 2004. № 2. С.113–117.
2. Шевченко В.Н. Характеристические многочлены многоиндексных транспортных задач. //Дискретная математика Т.15. Вып. 2. 2003. С. 83–88.
3. Титова Е.Б., Шевченко В.Н. О минорах матрицы ограничений многоиндексных транспортных задач. // Дискретная математика. 2012. Т. 24. Вып. 4. С. 147–157.
4. Шевченко В.Н. Качественные вопросы целочисленного линейного программирования. М.: Наука, 1995.

5. Мишина А.П., Проскуряков И.В. Высшая алгебра. Линейная алгебра, многочлены, общая алгебра. М.: Наука, 1965.
6. Титова Е.Б., Шевченко В.Н. Сингулярный многочлен матрицы инцидентности  $d$  – мерного куба: Материалы IX Международного семинара "Дискретная математика и ее приложения" (Москва, МГУ, 18-23 июня 2007г.). М.: Изд-во механико-математического факультета МГУ, 2007. С. 250–253.
7. Титова Е.Б. Исследование миноров матрицы инцидентности  $d$  – мерного куба: Российская конференция «Дискретная оптимизация и исследование операций»: Материалы конференции (Алтай, 27 июня – 3 июля 2010). Новосибирск: Изд-во Ин-та математики, 2010. С. 125.

# Применение метода Монте-Карло для вычисления интегралов

*Д.А. Торбеева*

*Поволжский Государственный Технологический Университет*

Важнейшим приемом построения методов Монте-Карло является сведение задачи к расчету математических ожиданий. Так как математические ожидания чаще всего представляют собой обычные интегралы, то центральное положение в теории метода Монте-Карло занимают методы вычисления интегралов.

*Ключевые слова:* метод Монте-Карло, интегралы, случайные числа, моделирование случайных величин.

## 1. Введение

Методами Монте-Карло называют численные методы решения математических задач при помощи моделирования случайных величин. Однако, решать методами Монте-Карло можно любые математические задачи, а не только задачи вероятностного происхождения, связанные со случайными величинами.

Метод Монте-Карло, необходимый в тех случаях, когда аналитическое вычисление невозможно, к примеру, кратного интеграла. Также, численные методы применяются тогда, когда первообразная функция настолько сложна, что легче и быстрее вычислить с помощью всё того же метода Монте-Карло. Метод Монте-Карло обладает следующими преимуществами: применим в многомерных случаях; лёгкость применения при небольших ограничениях.

Также данный метод имеет и свои недостатки. А именно: зависимость метода от случайных значений, которые могут распределиться по области интегрирования неравномерно; большая требовательность к ресурсам вычислительной техники при очень большом количестве случайных значений; границы ошибки не определены точно, но включают некую случайность.

Цель работы – изучение особенностей применения метода Монте-Карло для вычисления интегралов.

Для достижения цели были поставлены следующие задачи:

- 1) изучение теоретических основ метода Монте-Карло;
- 2) определить для каких интегралов лучше всего использовать метод Монте-Карло;
- 3) рассмотреть задачи на этот метод.

## 2. Описание метода Монте-Карло

### 2.1. История метода Монте – Карло

Метод Монте-Карло можно определить, как метод моделирования случайных величин с целью вычисления характеристик их распределений. Определение метода можно сделать более общим, говоря не о случайных величинах, а о случайных функциях.

Идея моделирования случайных явлений известна давно и, по мнению некоторых авторов, восходит ко временам Древнего Вавилона и Ветхого Завета. Что же касается использования такого рода явлений для целей приближенных вычислений, то первой работой в этой области принято считать работу Холла 1873 г. о вычислении числа  $\pi$  с помощью случайных бросаний иглы на разграфленную параллельными линиями бумагу. Существуют также ряд более поздних работ, в которых до появления электронных вычислительных машин (ЭВМ) использовались по существу идеи метода Монте-Карло. До появления электронно-вычислительных машин (ЭВМ) этот метод не мог получить широкого распространения, так как ручное моделирование случайных величин является очень трудоемкой и изнурительной работой. Поэтому, идеи эти не

получили заметного развития вплоть до 1944 г., когда в связи с работами по созданию атомной бомбы Дж. фон Нейман предложил широко использовать аппарат теории вероятностей для решения прикладных задач с помощью ЭВМ. Первая работа, где этот вопрос систематически излагался, принадлежит Метрополису и Уламу.

В Советском Союзе первые статьи о методе Монте-Карло были опубликованы в 1955–1956 годах (Чавчанидзе, Шрейдер). В настоящее время имеется более двух тысяч работ, где исследуются теоретические основы метода или рассматриваются его применения к конкретным задачам. Метод используется для решения прикладных задач из большого числа областей науки и техники. Первоначально метод Монте-Карло использовался главным образом для решения задач нейтронной физики, где традиционные численные методы оказались малопригодными. Далее его влияние распространилось на широкий круг задач статистической физики, очень разных по своему содержанию. К разделам науки, где все в большей мере используется метод Монте-Карло, следует отнести задачи теории массового обслуживания, задачи теории игр и математической экономики, задачи теории передачи сообщений при наличии помех и ряд других. Метод Монте-Карло оказал и продолжает оказывать существенное влияние на развитие методов вычислительной математики (например, развитие методов численного интегрирования) и при решении многих задач успешно сочетается с другими вычислительными методами и дополняет их.

Его применение оправдано в первую очередь в тех задачах, которые допускают теоретико-вероятностное описание. Это объясняется как естественностью получения ответа с некоторой заданной вероятностью в задачах с вероятностным содержанием, так и существенным упрощением процедуры решения.

## 2.2. Метод Монте – Карло

Важнейший прием построения методов Монте-Карло – сведение задачи к расчету математических ожиданий. Пусть требуется найти значение  $m$  некоторой изучаемой величины. С этой целью выбирают такую случайную величину  $X$ , математическое ожидание которой равно  $m$ :  $M[X] = m$ . Практически же поступают так: вычисляют (разыгрывают)  $N$  возможных значений  $x_i$  случайной величины  $X$ , находят их среднее арифметическое  $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ . Так как последовательность одинаково распределенных случайных величин, у которых существуют математические ожидания, подчиняется закону больших чисел, то при  $N \rightarrow \infty$  среднее арифметическое этих величин сходится по вероятности к математическому ожиданию. Таким образом, при больших  $N$  величина  $\bar{x} \approx m$ .

В методе Монте-Карло данные вырабатываются искусственно путем использования некоторого генератора случайных чисел в сочетании с функцией распределения вероятностей для исследуемого процесса. Таким генератором может быть таблица, колесо рулетки, подпрограмма ЭВМ или какой-либо другой источник равномерно распределенных случайных чисел. Подлежащее разыгрыванию распределение вероятностей может быть основано на эмпирических данных, извлекаемых из ранее сформированных записей, или на результатах последнего эксперимента либо может представлять собой известное теоретическое распределение.

Таким образом, для применения метода Монте-Карло необходимо уметь разыгрывать случайную величину.

Приведем так называемые правила для разыгрывания случайных величин.

Введем обозначения:  $R$  – непрерывная случайная величина, распределенная равномерно в интервале  $(0, 1)$ ;  $r_i$  ( $i = 1, 2, \dots$ ) – случайные числа (возможные значения  $R$ ).

**Правило 1.** Для того, чтобы разыграть дискретную случайную величину  $X$ , заданную законом распределения (см. табл. 1), следует разбить интервал  $(0, 1)$  оси  $Or$  на  $n$  частичных интервалов.

**Таблица 1.** Распределение значений

$X$	$x_1$	$x_2$	...	$x_n$
$p$	$p_1$	$p_2$	...	$p_n$

Выбрать (например, из таблицы случайных чисел) случайное число  $r_i$ . Если  $r_i$  попало в частичный интервал, то разыгрываемая величина приняла возможное значение  $x_i$ .

**Правило 2.** (Метод обратных функций). Для того чтобы разыграть возможное значение  $x_i$  непрерывной случайной величины  $X$ , зная ее функцию распределения  $F(x)$ , надо выбрать случайное число  $x_i$ , приравнять его функции распределения и решить относительно  $x_i$  полученное уравнение  $F(x_i) = r_i$ .

**Правило 3.** Для того, чтобы разыграть возможное значение  $x_i$  непрерывной случайной величины  $X$ , зная ее плотность вероятности  $f(x)$ , надо выбрать случайное число  $r_i$  и решить относительно  $x_i$  уравнение

$$\int_{-\infty}^{x_i} f(x)dx = r_i,$$

или уравнение

$$\int_a^{x_i} f(x)dx = r_i.$$

где  $a$  – наименьшее конечное возможное значение  $X$ .

### 3. Вычисление интегралов

#### 3.1. Алгоритм вычисления интегралов методом Монте-Карло

1. Задаем область интегрирования с помощью неравенств; параметр  $t_\beta$ , соответствующий коэффициенту доверия  $\beta$ ; подынтегральную функцию  $f(x_1, \dots, x_m)$ ; количество испытаний  $N$ .
2. Подбираем такие  $a_i$  и  $b_i$ , что  $a_i \leq x_i \leq b_i$  ( $i = \overline{1, m}$ ), т.е. задаем векторы  $\vec{a} = [a_1, \dots, a_m]$  и  $\vec{b} = [b_1, \dots, b_m]$ . Вычисляем объем параллелепипеда, ограничивающего область интегрирования, по формуле  $V = (b_1 - a_1) \cdot \dots \cdot (b_m - a_m)$ .
3. Фиксируем  $n = 0, sumF = 0, sumF2 = 0$ .
4. Для  $k$  от 1 до  $N$ :
  - а) генерируем случайные точки  $(\xi_1^{(k)}, \dots, \xi_m^{(k)})$ , такие что  $0 \leq \xi_1^{(k)} \leq 1, i = \overline{1, m}$ ;
  - б) вычисляем  $x_i^{(k)} = a_i + (b_i - a_i) \xi_i^{(k)}$   $i = \overline{1, m}$ ;
  - в) если для  $x^{(k)} = (x_1^{(k)}, \dots, x_m^{(k)})$  выполняются неравенства, задающие область интегрирования, то
 
$$fx = f(x^{(k)})$$

$$n = n + 1;$$

$$sumF = sumF + fx;$$

$$sumF2 = sumF2 + fx^2;$$

5. Вычисляем  $\overline{F}_{cp} = \frac{1}{n} \text{sum} F$ ,  $\overline{F^2}_{cp} = \frac{1}{n} \text{sum} F^2$ ,  $\overline{\Omega} = \frac{n}{N}$ .
  6. Вычисляем значение интеграла по формуле  $\bar{I} = V \cdot \overline{F}_{cp} \cdot \overline{\Omega}$ .
  7. Находим приближенные значения дисперсий  $S_1^2 = \overline{F^2}_{cp} - (\overline{F}_{cp})^2$ ,  $S_2^2 = \overline{\Omega} \cdot (1 - \overline{\Omega})$ .
  8. Вычисляем значение погрешности  $\Delta = V \cdot t_\beta \cdot \frac{\overline{\Omega} \cdot S_1}{\sqrt{n}} + \frac{|\overline{F}_{cp}| \cdot S_2}{\sqrt{N}}$ .
  9. Если погрешность  $\Delta$  – велика, то задаем новую погрешность  $\varepsilon$  и вычисляем  $N'$  – ориентировочное количество испытаний, необходимое для достижения указанной точности  $\varepsilon$ , по формуле
  10.  $N' = \left[ \left( \frac{c^2}{\varepsilon} \right) \right] + 1$ , где  $c = V \cdot t_\beta \cdot (S_1 \sqrt{\overline{\Omega}} + S_2 \cdot |\overline{F}_{cp}|)$ ,  $[ ]$  – означает целую часть числа.
- Выполняем пункты 3–7 с полученным необходимым количеством испытаний  $N_0$ .

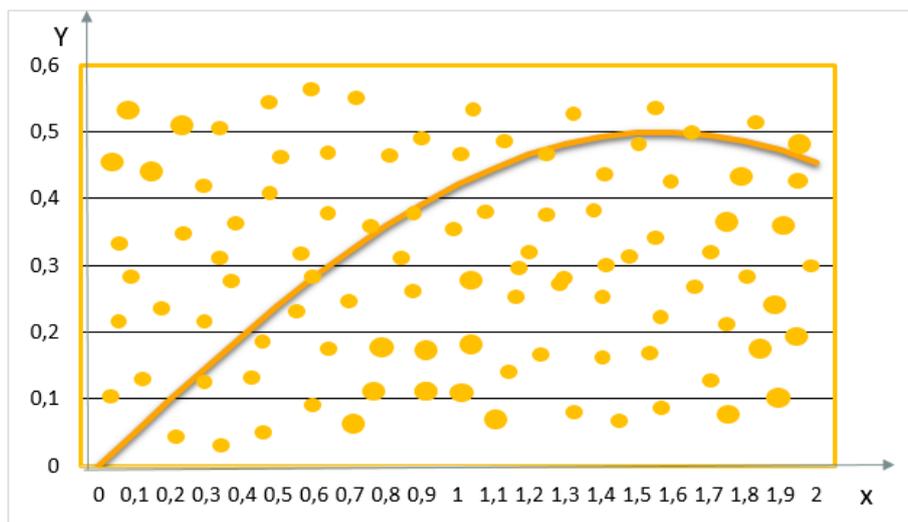
### 3.2. Геометрический алгоритм решения интегралов

Пусть область интегрирования  $[a, b]$  делится случайным образом на участки шириной  $(a - b)/N$ , где  $N$  – количество случайных точек, и затем площадь получившихся столбиков суммируется.

Нужно взять область интегрирования  $[a, b]$ , ограничить ее прямоугольником с площадью  $S_{par}$  и набросать в этот прямоугольник случайным образом точек. Затем требуется посчитать количество точек  $K$ , попавших в область под графиком функции и вычислить интеграл (площадь под кривой  $S$ ) по формуле:

$$S = S_{par} \frac{K}{N}.$$

**Пример 1.** Найти этим методом интеграл функции  $f(x) = \frac{\sin(x)}{2}$  на отрезке  $[0, 2]$ .



**Рис. 1.** График функции  $y = \frac{\sin(x)}{2}$

Решим интеграл традиционным способом:

$$\int_0^2 \frac{\sin(x)}{2} dx = \frac{1}{2} \int_0^2 \sin(x) dx = -\frac{1}{2} \cos(x) \Big|_0^2 = -\frac{1}{2} [\cos 2 - \cos 0] = -\frac{1}{2} [\cos 2 - 1] \approx 0,70807.$$

Ограничиваем область интегрирования прямоугольником размером  $2 \times 1,5$  и вбрасываем сотню точек со случайными координатами. Повторяем этот эксперимент 20 раз. На рис. 1 показан один из 20 результатов.

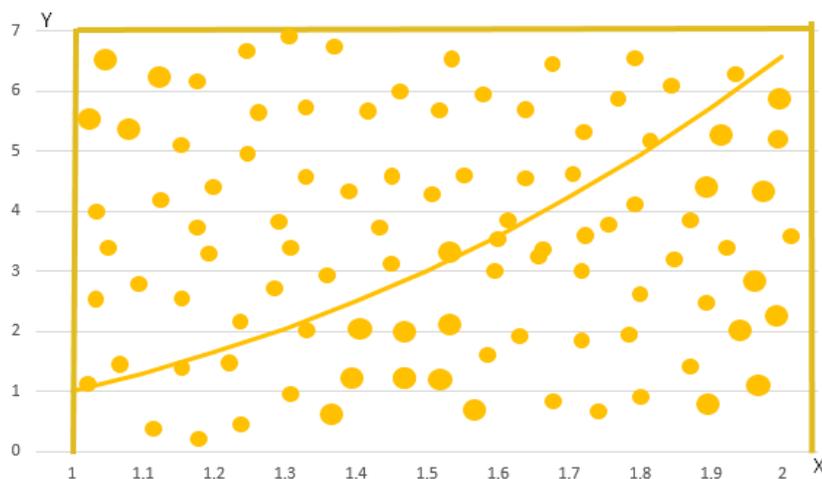
При таком малом количестве точек значение предсказуемо бессовестно скачет от 0.51 до 0.99, со средним 0.735 и такой же медианой. После увеличиваем количество точек в сто раз до 10000. Разброс значений теперь становится значительно меньше: между 0.6849 и 0.732, со средним 0.705825 и медианой 0.7077. Увеличиваем количество точек еще в сто раз до миллиона. Миллион точек на изображении превращается в сплошную заливку прямоугольника. Разброс значений оказывается между 0.705135 и 0.709499, со средним 0.70764705 и медианой 0.7075275. Последняя чередя экспериментов со ста миллионами точек. Точность результатов повышается, диапазон значений оказывается между 0.70788846 и 0.70843821. Среднее значение 0.708100675 и медиана 0.70808943.

**Пример 2.** Вычислить интеграл  $\int_1^2 e^x dx$ .

Решим интеграл традиционным способом:

$$\int_1^2 e^x dx = e^x \Big|_1^2 = e^2 - e^1 \approx 4,67077.$$

Рисуем график функции  $y = e^x$  на отрезке  $[1; 2]$  (см. рис. 2).



**Рис. 2.** График функции  $y = e^x$

Вбрасываем случайным образом сначала 10 точек, после увеличиваем это количество до 100. Результаты приведены в таблице 2.

**Таблица 2.** Интегрирование методом Монте-Карло

Число точек	Метод Монте-Карло
10	4.62289422
100	4.69812790

**Пример 3.** Вычислить интеграл  $\int_0^2 xe^{-x} dx$ .

Решим интеграл традиционным способом:

$$\begin{aligned} \int_0^2 xe^{-x} dx &= |u = x, dv = e^{-x}, du = dx, v = \int dv = \int e^{-x} = -e^{-x}| \\ &= -xe^{-x} \Big|_0^2 - \int_0^2 -e^{-x} dx = (-xe^{-x} - e^{-x}) \Big|_0^2 = (-2e^{-2} - e^{-2}) - (-0e^0 - e^0) \\ &= -3e^{-2} + 1 = 1 - 3e^{-2} \approx 0,593994. \end{aligned}$$

Рисуем график функции  $y = xe^{-x}$  на отрезке  $[0; 2]$  (см. рис. 3). Вбрасываем случайным образом сначала 10 точек, после увеличиваем это количество до 100. Результаты приведены в таблице 3.

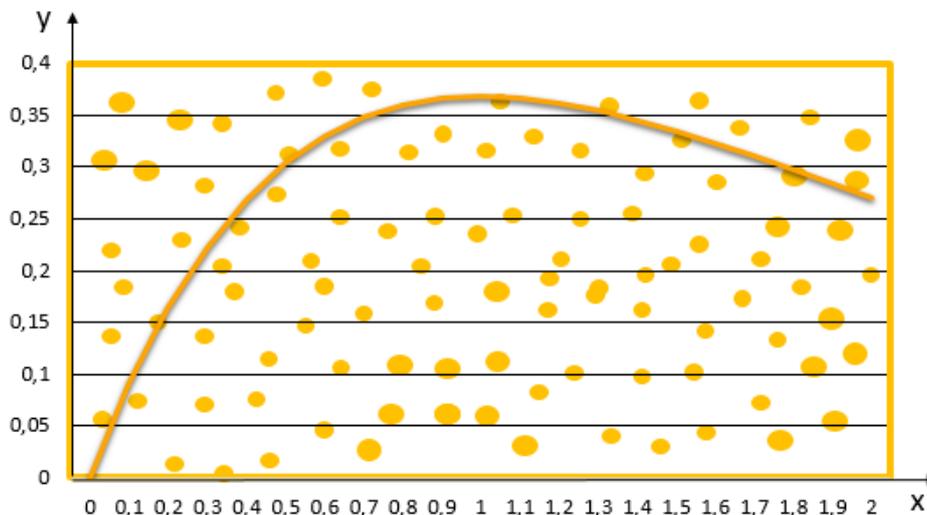


Рис. 3. График функции  $y = xe^{-x}$

Таблица 3. Интегрирование методом Монте-Карло

Число точек	Метод Монте-Карло
10	0.28397
100	0.582978

## Литература

1. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. 9-е изд. Москва: Лаборатория знаний, 2020. 636 с.
2. Данко П.Е. Высшая математика в упражнениях и задачах. В 2 частях Часть 2 / П.Е. Данко, А.Г. Попов, Т.Я. Кожевникова. 4-е изд.
3. Соболев И.М. Метод Монте-Карло. М.: Наука, 1968. 64 с.
4. [https://ru.wikipedia.org/wiki/Метод\\_Монте-Карло](https://ru.wikipedia.org/wiki/Метод_Монте-Карло).

# Асимптотика собственного значения оператора Шредингера с растущим потенциалом

*И.Х. Хуснуллин, З.И. Мухаметрахимова*

*Башкирский государственный педагогический университет им. М. Акмуллы*

В работе исследуется асимптотическое поведение собственных значений оператора Шредингера с растущим потенциалом при стремлении малого параметра к нулю. Для собственных значений и соответствующих собственных функций строятся первые члены асимптотических разложений.

*Ключевые слова:* оператор Шредингера, асимптотика, собственное значение, собственная функция, сингулярные задачи.

## 1. Введение

Асимптотические методы являются одним из наиболее мощных инструментов современной прикладной математики для решения различных задач. Они находят широкое применение в механике, физике и в других науках, оперирующих дифференциальными уравнениями. Это объясняется тем, что задачи, которые рассматриваются в теории дифференциальных уравнений, в подавляющем большинстве не имеют явного решения в виду сложной зависимости от числовых и функциональных параметров, входящих в эти задачи. Однако правильное описание решения или нахождение приближенного решения можно существенно упростить, если известно, что некоторые из параметров очень малы, либо, наоборот, велики. Для решения таких задач можно применить асимптотические методы. Эти методы, как правило, связаны со спецификой рассматриваемой задачи. Одним из классов задач, которые успешно решаются применением асимптотических методов, являются сингулярно возмущенные задачи. Такие задачи описывают многие реальные модели окружающего мира.

В данной работе рассматривается задача Дирихле на собственные значения оператора Шредингера с растущим потенциалом. Оператор Шредингера представляет собой оператор эллиптической сингулярной краевой задачи. Математическая теория операторов Шредингера используется в квантовой механике, дифференциальной геометрии, топологии.

Целью работы является выяснение асимптотического поведения собственных значений оператора Шредингера с растущим потенциалом. Заметим, что здесь рассматривается возмущение, которое носит сингулярный характер и поэтому при построении асимптотик будет использоваться метод согласования асимптотических разложений.

В работе построены первые члены асимптотических разложений собственных значений и собственных функций оператора Шредингера с растущим потенциалом по малому параметру.

## 2. Постановка задачи и формулировка результатов

### 2.1. Постановка задачи

Рассматривается следующая задача Дирихле на собственные значения:

$$-y_\varepsilon'' + \ln \varepsilon V\left(\frac{x-x_0}{\varepsilon}\right)y_\varepsilon = \lambda_\varepsilon y_\varepsilon, \quad x \in (0,1), \quad (1)$$

$$y_\varepsilon(0) = y_\varepsilon(1) = 0, \quad (2)$$

где  $\lambda_\varepsilon$  – собственное значение;  $y_\varepsilon$  – соответствующая собственная функция;

$V \in C_0^\infty(R)$  (т. е.  $V$  – финитная бесконечно дифференцируемая функция носитель которой содержится в интервале  $(0,1)$ );

$x_0 \in (0,1)$ ;

$\varepsilon$  – малый положительный параметр ( $0 < \varepsilon \ll 1$ ).

Требуется построить первые члены асимптотического разложения для собственной функции  $y_\varepsilon$  и собственного значения  $\lambda_\varepsilon$ .

Построение асимптотик будет проводиться методом согласования асимптотических разложений [3].

В работе [4] был рассмотрен случай, где в задаче (1) вместо  $\ln\varepsilon$  берется  $\varepsilon^{-\gamma}$ , где  $\gamma \in (0,1)$ .

Аналогично работам [1], [5] доказывается сходимости  $\lambda_\varepsilon \rightarrow \lambda_0$ ,  $y_\varepsilon \rightarrow y_0$ , где  $\lambda_0$  – собственное значение, а  $y_0$  – соответствующая нормированная в  $L_2(0,1)$ .  $\lambda_0$  и  $y_0$  – получаем переходя из уравнения (1) в предельное уравнение (т.е. уравнение без потенциала). Предполагается, что предельное уравнение имеет бесконечное число собственных значений [2].

## 2.2. Метод исследования

Нахождение точного аналитического выражения для решения краевых задач для дифференциального уравнения или уравнения в частных производных в большинстве случаев вызывает затруднение, а иногда даже становится невозможно.

Но довольно часто подобные задачи содержат некоторый малый параметр  $\varepsilon$ . Таким параметром в зависимости от рассматриваемой задачи может быть отношение некоторых физических величин, имеющих отношение к изучаемому явлению.

Суть метода согласования асимптотических разложений [3] заключается в следующем.

Пусть задача рассматривается в пространстве, внутри и вне некоторой ограниченной области. Во большинстве случаев для искомого решения  $u(x, \varepsilon)$  не существует единого равномерного всюду асимптотического приближения вида:

$$u(x, \varepsilon) = \sum_{k=1}^{\infty} u_k(x)g_k(\varepsilon). \quad (3)$$

Подходящим образом составляем внутреннее и внешнее асимптотические разложения, представляющее решение изучаемой задачи. Например, внешнее разложение берем в виде ряда (3), а внутреннее – в виде:

$$u(x, \varepsilon) = \sum_{k=1}^{\infty} v_k(\theta)h(\varepsilon), \quad (4)$$

где  $\theta = \theta(\varepsilon)$ .

Если их конструкция верна, то в пограничном слое эти разложения должны совпадать. И после такого согласования получаем асимптотический ряд, представляющий решение данной задачи в данном пространстве.

В задаче (1), (2) рассматривается возмущение, которое носит сингулярный характер и поэтому построение асимптотик будет проводится методом согласования асимптотических разложений.

## 2.3. Основные результаты

Применяя описанный метод к нашей задаче (1), (2) приходим к следующим результатам.

Асимптотика собственной функции  $y_\varepsilon$  состоит из внешнего и внутреннего разложения.

Внешнее разложение имеет следующий вид:

$$y_\varepsilon^+(x) = y_0(x) + \varepsilon \ln \varepsilon y_{1,1}^+(x) + \dots, \quad x > x_0, \quad (5)$$

$$y_\varepsilon^-(x) = y_0(x) + \varepsilon \ln \varepsilon y_{1,1}^-(x) + \dots, \quad x < x_0, \quad (6)$$

где

$$y_0(x) = \sqrt{2} \sin(\sqrt{\lambda_0}x), \quad (7)$$

$$y_{1,1}^-(x) = \frac{y_0(x_0) \int_{-\infty}^{+\infty} V(t)dt}{2\lambda_0} (y_0(x_0)y_0'(x)x - y_0'(x_0)y_0(x)x_0), \quad (8)$$

$$y_{1,1}^+(x) = \frac{y_0(x_0) \int_{-\infty}^{+\infty} V(t)dt}{2\lambda_0} (y_0(x_0)y_0'(x)(x-1) - y_0'(x_0)y_0(x)(x_0-1)). \quad (9)$$

Внутреннее разложение имеет следующий вид:

$$y_\varepsilon^{in}(\xi) = v_0(\xi) + \varepsilon v_1(\xi) + \varepsilon^2 \ln \varepsilon v_{1,1}(\xi) + \dots, \quad (10)$$

где

$$v_0(\xi) = y_0(x_0), \quad (11)$$

$$v_1(\xi) = y_0'(x_0)\xi + a_{1,1}, \quad (12)$$

$$v_{1,1}(\xi) = y_0(x_0) \int_{-\infty}^{\xi} \int_{-\infty}^{\tau} V(t) dt d\tau + b_{1,1}\xi + a_{1,1}, \quad (13)$$

$b_{1,1}$ ,  $a_{1,1}$  – произвольные константы, которые определяются при построение следующих членов асимптотики.

Во внутреннем разложении  $\xi = \frac{(x-x_0)}{\varepsilon}$  – растянутые переменные.

Асимптотика собственного значения  $\lambda_\varepsilon$  имеет вид:

$$\lambda_\varepsilon = \lambda_0 + \varepsilon \ln \varepsilon \lambda_{1,1} + \dots, \quad (14)$$

где:

$$\lambda_0 = (\pi n)^2, \quad n \in N, \quad (15)$$

$$\lambda_{1,1} = y_0^2(x_0) \int_{-\infty}^{+\infty} V(t) dt. \quad (16)$$

### 3. Заключение и выводы

Таким образом, в данной работе была построена двучленная формальная асимптотика собственного значения  $\lambda_\varepsilon$  и соответствующей собственной функции  $y_\varepsilon$  оператора Шредингера с растущим потенциалом при стремлении малого параметра к нулю. Построение асимптотик проводилось методом согласования асимптотических разложений.

### Литература

1. Гадыльшин Р.Р. Асимптотика собственного значения с сингулярно возмущенной эллиптической задачи с малым параметром в граничном условии // Дифференциальные уравнения. 1986. № 4. С. 640–652.
2. Егоров Ю.В., Шубин М.А. Линейные дифференциальные уравнения с частными производными. Основы классической теории // Итоги науки и техники. Серия «Современные проблемы математики. Фундаментальные направления». М.: ВИНТИ. Т. 31. 1988.
3. Ильин А.М., Данилин А.Р. Асимптотические методы в анализе. М.: ФИЗМАТЛИТ, 2009. 248 с.
4. Хуснуллин И. Х. Двучленная формальная асимптотика собственного значения оператора Шредингера с растущим потенциалом, локализованным на сжимающемся множестве // Ученые записки сборник научных статей физико-математического факультета. Уфа: Башкирский государственный педагогический университет им. М. Акмуллы, 2007. С. 26–31.
5. Планида М.Ю. О сходимости решений сингулярно возмущенных краевых задач для лапласиана // Мат. заметки. 2001. № 6. С. 867–877.

# Аппроксимация картографических объектов территорий, пострадавших от стихийных бедствий

*К.А. Числова, Е.В. Борисова*

*Тверской Государственный Технический университет*

В статье рассмотрен вопрос разработки алгоритма определения эффективной детализации пораженной территории. Основным методом выбрано моделирование геометрических объектов векторным способом. Выполненные модельные расчеты показали достаточную точность аппроксимации и перспективы построения интеллектуальных инструментов. Исследовательские работы студентов младших курсов в области экологического образования способствуют освоению фундаментальных, прикладных знаний по математике и специальным дисциплинам.

*Ключевые слова:* экологические проблемы, моделирование, приближенные оценки.

Необходимость поддержания целей устойчивого развития для сохранения экосферы нашей планеты, требуют нового подхода к разрешению проблем в области экологического образования, в широком смысле этого понятия. Для вывода экологического мышления студентов инженерных вузов на новый уровень, формирования высокой эколого-правовой культуры и ответственного отношения бизнеса и производства к обществу, человеку и окружающей среде, необходимо обеспечить фундаментальность образования и саморазвитие, не противоречащие динамике движения человечества.

Несмотря на принятые в разные годы законы и постановления Российской Федерации: «Об экологическом образовании обучающихся в образовательных учреждениях РФ» (1994); «Об охране окружающей среды» (2002); «Паспорт национального проекта «Экология» (2018), неудовлетворенность общества в вопросах разрешения экологических проблем не снижается. Среди развивающихся инженерных направлений подготовки в вузах России существенное место занимает явное или опосредованное экологическое образование. Инертность и консерватизм мышления людей, нацеленность на прибыли и излишнее потребление «выключает» осознание долгосрочных последствий от ухудшения состояния окружающей среды, истощения природных ресурсов. В современном мире существует множество экологических рисков, которые необходимо учитывать. Существенный сдвиг экобаланса на планете происходит под влиянием участвовавших лесных пожаров, разрушительных смерчей, цунами, наводнений и других природных явлений. Однако, во многих случаях, именно люди несут ответственность за последствия нарушения технологии, халатности или стремления бизнеса получить сиюминутную выгоду.

В Тверском государственном университете более 20 лет существует факультет природопользования и инженерной экологии. Здесь, среди других специальностей, готовят специалистов по профилю «Технические средства природообустройства и защиты в чрезвычайных ситуациях». Выполнение расчетно-исследовательских работ на младших курсах способствует освоению фундаментальных и прикладных знаний по математике, а также является основой для изучения специальных дисциплин профессионального цикла.

Территория России регулярно подвергается воздействиям стихийных бедствий и чрезвычайных ситуаций. Основными из них являются сезонные наводнения и лесные пожары. Причинами наводнений служат быстрое таяние большого объема снега, образующиеся на реках заторы из льдин. Основным фактором затоплений в теплое время являются сильные продолжительные дожди. Именно так произошло в конце лета 2013 года на Дальнем Востоке, где проливные ливни вызвали самое масштабное наводнение за последние 115 лет [1].

Большинство пожаров возникает в результате сжигания мусора, сельскохозяйственных палов, в местах пикников, от брошенной горящей спички. В июне 2021 года таёжные леса в Сибири и на Дальнем Востоке охватили лесные пожары, последовавшие за рекордной жарой и засухой. К 16 августа 2021 года выгорело более 17 млн гектаров [2]. По некоторым источникам, это больше, чем все другие пожары в мире вместе взятые, как минимум со времен первых спутниковых наблюдений за Землей.

Прогнозирование и предотвращение экологического и экономического ущерба производится средствами мониторинга территорий как наземными, так и со спутников. Для этого в России действуют профессиональные службы сбора, структурирования и анализа данных. В службах МЧС России профессиональные беспилотные летательные аппараты применяют в поисково-спасательных работах, в наблюдении за ледовой и паводковой обстановкой, выявлении очагов лесных пожаров. Для решения подобных задач выполняется целевое картографирование. На его основе можно выявлять территориальные границы чрезвычайной ситуации, а также динамически обновлять карту событий. Информацию важно получать как при работе с небольшими возгораниями для установки точного месторасположения и направления распространения, так и с крупномасштабными пожарами для очерчивания границ чрезвычайной ситуации и контроля вектора ее распространения. Полученные карты-снимки территорий представляют информацию для экспресс оценки пораженных площадей. Такие оценки не требуют слишком высокой точности получаемых результатов и могут использовать приближенные методы.

Рассмотрим метод аппроксимации границ картографических объектов линейными функциями. В широком смысле аппроксимация или приближение заключается в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми [3,4].

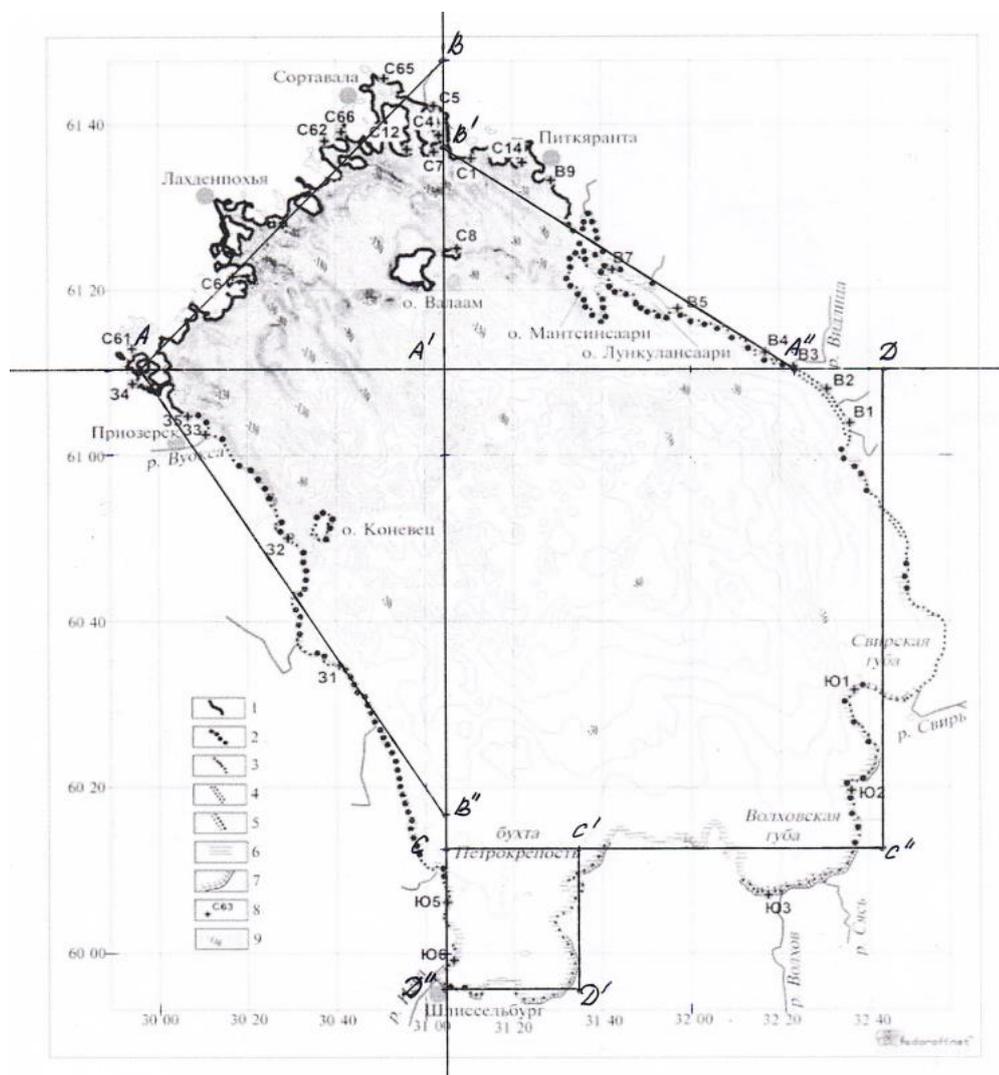
Изучение различных литературных источников показало, что в такие методы используется либо сложная математика, либо закрытые специальные продукты и алгоритмы. Сформулируем исследовательскую и расчетную задачу: аппроксимировать территориальные границы и внутренние области картографических объектов элементарными геометрическими формами с использованием векторного способа. Требуется определить максимально эффективную детализацию измеряемой территории, с сохранением ее особенностей, не перегружая деталями.

Карты представляются в различных форматах, с некоторым масштабом, поэтому выполнялась предварительная обработка данных. Этот параметр напрямую влияет на количество элементарных площадей, которые возможно получить. Чем выше точность карты, тем больше приходится обрабатывать данных. Координаты могут быть как географическими, так и линейными. После предварительной обработки получаем данные в унифицированном формате. Линии называются границами, если при их пересечении происходит существенное изменение одного или многих атрибутов местности. Другими словами, важность границ обусловлена их функцией по отношению к прилежащим территориям [5]. В контексте поставленной задачи – граница разделяет пораженную и непораженную территории. Вычисление площадей можно проводить по данным, представленным растровым и векторным способом. В нашей работе использован метод векторного представления. Ограничивающая ломаная линия состоит из отдельных прямых, построенных на основе векторов по координатам двух точек. Полученная граница должна быть замкнутой и пересечения линий не допускаются.

Работа проведена в три этапа: подготовка данных; построение граничной линии и элементарных площадей, средствами векторного представления оцениваемой территории; построение аппроксимирующего решения, состоящего в суммировании несколько простых фигур, площади которых легко определяются по известным формулам.

В качестве модельного примера рассчитаем площадь водного зеркала Ладожского озера, аппроксимируя фрактальную береговую линию отрезками прямых и построив расчетный многоугольник. Так как площадь зеркальной поверхности известна, можно по результатам аппроксимации оценить точность приближенного метода и сделать вывод о его пригодности для построения алгоритмов экспресс оценок.

Топографическая карта Ладожского озера представлена на рис.1, информация взята из открытого источника «Карты Ладожского озера» [6].



**Рис 1.** Топографическая карта Ладожского озера с нанесенными аппроксимирующими линиями

На первом этапе необходимо подготовить картографические данные, чтобы снизить погрешности при преобразованиях. Выбираем точки, которые могут быть связаны прямой линией, наилучшим образом «спрямляющей» естественную границу. (Рисунок 1). Применяется достаточно простой подход к обработке исходных данных. Предположим, в некоторой части карты у нас встречается несколько точек. Две из них (опорные) соединены и представляют прямую и есть еще точки расположенные ближе или дальше от построенной линии. Чтобы избежать значительных искажений в покрытии картографического объекта, границы элементарных площадей максимально приближены к опорным точкам.

С помощью градусной сетки определим величину нужных линий. Известно, что все меридианы имеют одинаковые длины, следовательно, длина всех дуг величиной  $1^\circ$  составляет

111,3 км. Длины параллелей величиной 1° различаются, она увеличивается от полюса к экватору [7] (см. табл. 1).

**Таблица 1.** Длины используемых широт

Широта	Длина 1°, км
60°	55,8
61°	54,1

Использованы формулы:

$$A = (d_1 \pm d_2) \times 55,8;$$

$$B = (ш_1 \pm ш_2) \times 111,3,$$

где  $d_1, d_2$  – долгота;  $ш_1, ш_2$  – широта. Получим:

$$AA' = (31,067 - 29,923) \times 54,1 = 61,89;$$

$$A'B = (61,788 - 61,168) \times 111,3 = 69.$$

Используя картографические данные, построим таблицу 2, в которую внесем исходные координаты опорных точек, векторы и их длины (модули) для линий внешней границы и внутренних элементарных областей. В табл. 2 обозначены «N» – северная широта; «E» – восточная долгота.

**Таблица 2.** Картографические данные и расчетные значения

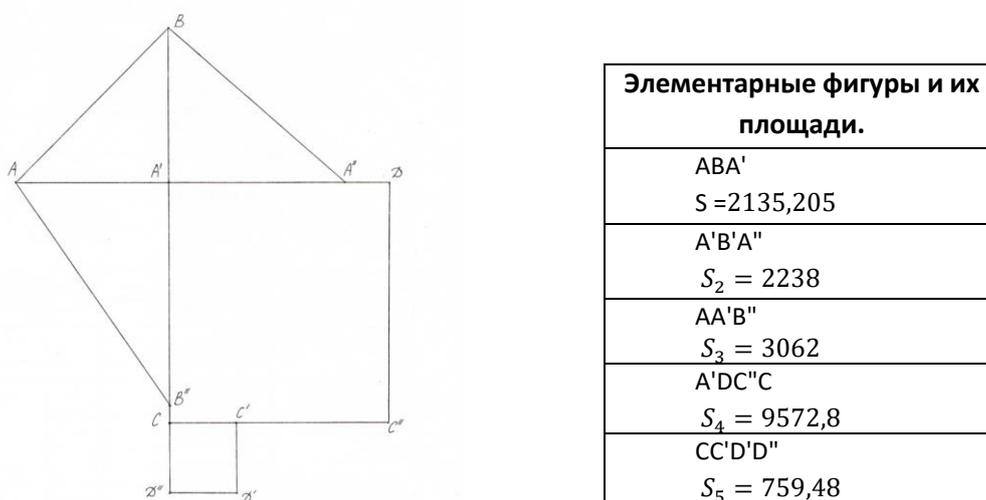
Координаты точек	Модули векторов границ внешних границ и внутренних областей
A(61,168 N; 29,923 E); B(61,788 N; 31,067 E); A'(61,168 N; 31,067 E)	$AA' = 61,89$ $A'B = 69$
A'(61,168 N; 31,067 E); B'(61,615 N; 31,067 E); A''(61,168 N; 32,394 E)	$A'B' = 49,75$ $A'A'' = 89,97$
A(61,168 N; 29,923 E); A'(61,168 N; 31,067 E); B''(60,279 N; 31,067 E)	$AA' = 61,89$ $A'B'' = 98,95$
A'(61,168 N; 31,067 E); D(61,168 N; 32,73 E); C''(60,212 N; 32,73 E); C(60,212 N; 31,067 E)	$A'D = 89,97$ $A'C = 106,4$
C(60,212 N; 31,067 E); C'(60,212 N; 31,577 E); D'(59,933 N; 31,577 E); D''(59,933 N; 31,067 E)	$CC' = 24,46$ $CD'' = 31,05$

Из табл. 2 видно, сколько различных данных необходимо обработать, чтобы получить адекватный результат. После предварительной обработки координат и создания таблицы данных на втором этапе строим элементарные площади, покрывающие оцениваемую территорию. Далее выполняются расчеты по всем подготовленным точкам для определения элементарных площадей.

Расчеты площадей, ограниченных участками прямых, с использованием подготовленных данных по координатам из табл. 2, обеспечивающих аппроксимацию водного зеркала озера, проведены по известным геометрическим формулам.

Например, площадь треугольника  $AA'B$   $S_1 = \frac{1}{2} \times AA' \times A'B$ . Площадь четырехугольника  $A'DC''C$   $S_4 = A'D \times A'C$ . Полученные результаты сведены в таблицу, включенную в рис. 2.

Графически результат аппроксимации элементарными фигурами с использованием векторного метода выглядит так, как показано на рис.2



**Рис. 2.** Аппроксимация водного зеркала Ладожского озера элементарными фигурами с расчетом их площадей

Третий этап. Завершает работу итоговый расчет в форме алгебраического суммирования. Вычислим аппроксимирующую площадь по выделенным фигурам:

$$\Sigma S = 2135,205 + 2238 + 3062 + 9572,8 + 579,48 = 17767,49 \text{ км}^2.$$

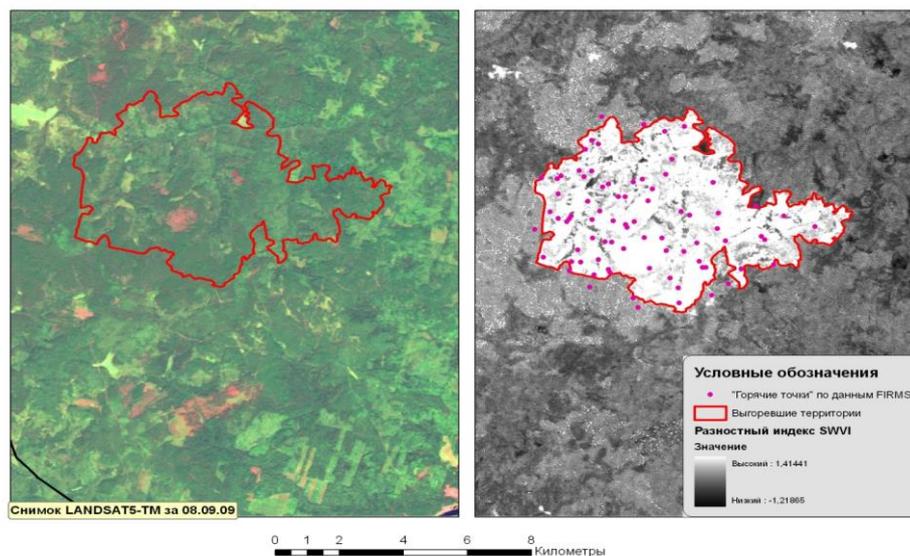
Найдем величину погрешности в результате применения методики аппроксимации, учитывая, что реальная площадь водного зеркала Ладожского озера по данным открытого источника [8] составляет  $S_{ис} = 17870$ . Вычислим абсолютную и относительную погрешности:

$$\delta = 17870 - 17767,49 = 102,51 \text{ км}^2$$

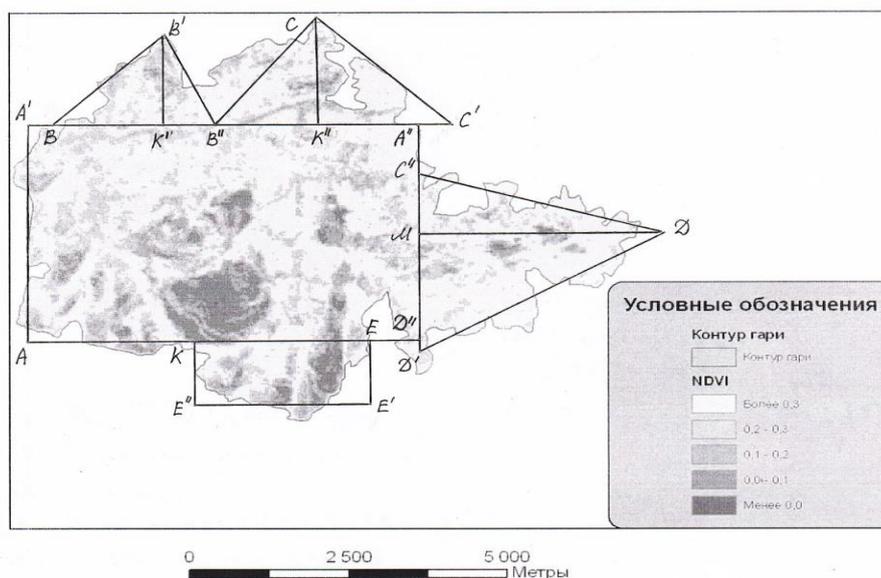
$$\Delta = \frac{102,51}{17870} = 0,0057$$

Предложенный и апробированный на известном объекте метод обеспечивает достаточно высокую точность оценки измеряемой площади. Можно говорить о его пригодности для построения алгоритмов экспресс оценок площадей, пораженных чрезвычайными ситуациями территорий.

Применим рассмотренную методику для оценки площади гари, образовавшейся в результате лесного пожара в Гайнском районе Пермского края 2010 года. На рис.3 показан картографический объект, изображение которого взято из открытого источника «Географические информационные системы и дистанционные зондирование» [9].



**Рис.3.** Картографические снимки гари в Гайнском районе Пермского края



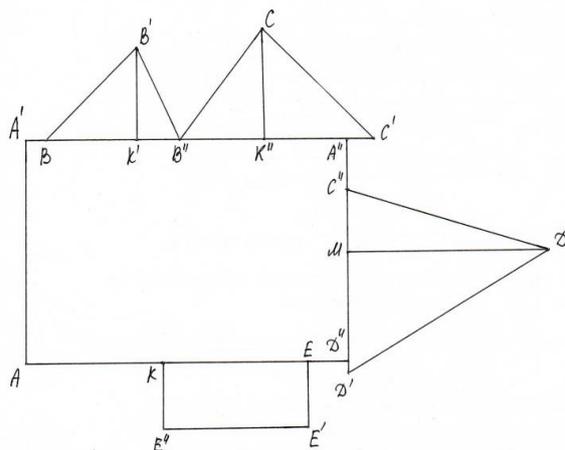
**Рис. 4.** Результат аппроксимации снимка гари элементарными фигурами

Расчеты площадей, ограниченных участками прямых, с использованием подготовленных данных по координатам, обеспечивающих аппроксимацию территории гари, проведены аналогично модельному примеру. Получены аппроксимирующие линии (векторы) для дальнейшего вычисления площадей.

**Таблица 3.** Аппроксимирующие линии

$AA'$	$AD''$	$BB'$	$B'K'$	$B''C'$	$CK''$	$C''D'$	$DM$	$KE''$	$E'E''$
3928,6	6142,86	2571,43	1642,86	3714,29	1785,71	3214,29	3857,14	1142,86	2714,29

Итоговые результаты предварительных расчетов представлены на рис.5.



Элементарные фигуры и их площади
$AA'A''D''$ $S_1 = 24,13$
$BB'B''$ $S_2 = 2,11$
$B''CC'$ $S_3 = 3,32$
$C''DD'$ $S_4 = 6,2$
$KEE'E''$ $S_5 = 3,1$

**Рис. 5.** Аппроксимация лесной гари Пермского края элементарными фигурами с расчетом их площадей

Вычислим суммарную (аппроксимирующую) площадь выделенных фигур:

$$\sum S = 24,13 + 2,11 + 3,32 + 6,2 + 3,1 = 38,86 \text{ км}^2.$$

С учетом погрешности метода составляющую 2%, получим экспресс оценку пораженной пожаром территории (сравнимую с данными МЧС [9]):  $S = 38,86 \pm 0,78$ .

Рассмотренный подход дает один из вариантов к решению поставленной задачи. Возможность перемещаться по карте с любым набором аппроксимирующих элементарных площадей открывает перспективы по разработке интеллектуальных алгоритмов динамического изменения детализации территорий.

## Литература

1. Крупнейшее наводнение 2013 года на Дальнем Востоке <https://23.mchs.gov.ru/deyatelnost/press-centr/novosti/4210243>
2. Лесные пожары в России [https://ru.wikipedia.org/wiki/Лесные\\_пожары\\_в\\_России\\_\(2021\)](https://ru.wikipedia.org/wiki/Лесные_пожары_в_России_(2021))
3. Бердышев В.И., Петрак Л.В. Аппроксимация функций, сжатие численной информации. Екатеринбург: УрО РАН, 1999. 296 с.
4. Курс лекций «Геоинформационные системы» / <https://kpfu.ru> > staff\_files
5. Карты Ладожского озера <https://fedoroff.net/load/maps/karta/obzornaja-/90-1-0-161>
6. Измерение расстояний с помощью градусной сетки <https://www.yaklass.ru/p/geografiya/5-klass/>
7. Ладожское озеро <https://iskatel.com/places/ladozhskoe-ozero>
8. Географические информационные системы и дистанционные зондирование <https://gis-lab.info/qa/fires-perm.html>
9. Оценка лесных пожаров в Пермском крае [GIS-Lab: Оценка последствий лесных пожаров 2010 г. в Пермском крае](#)

## Содержание

Предисловие .....	3
<b><i>В.А Беднаж, Д.С. Ермакова</i></b> Свойства классов $A\alpha^\infty$ и $X\alpha^\infty$ , аналитических в полуплоскости функций .....	5
<b><i>А.В. Боброва</i></b> Синхронизация волновых процессов в нейросетевых системах: возможности квантовых алгоритмов оценки фазы .....	11
<b><i>Д.С. Братухин</i></b> Улучшение стратегии управления рисками в системах с многоуровневой структурой .....	16
<b><i>Р.М. Вольников</i></b> Оптимизация аэродинамического профиля. Крыло с изменяемым профилем .....	18
<b><i>А.Т. Герюгова</i></b> Автоматизация подготовки рецепта на медицинские изделия .....	22
<b><i>Г.И. Захарычев</i></b> Приложение, рассчитывающее абсолютные и относительные показатели качества программного средства на основе иерархической модели, приведенной в ГОСТ 28195-99 .....	29
<b><i>А.Х. Исмадинов</i></b> Исследование возможностей API OpenAI .....	38
<b><i>И.А. Кириллова</i></b> Золотое сечение в фотографии.....	44
<b><i>Т.В. Кузнецова</i></b> Прикладное применение математики в программировании .....	48
<b><i>Р.В. Майтак, П.А. Пылов, А.В. Дягилева</i></b> Применение имитационной модели искусственного интеллекта для автоматизированной оптимизации процесса разработки угольных месторождений .....	52
<b><i>Д.М. Моисеев</i></b> Разработка мобильного приложения для диагностики и лечения компьютерной зависимости .	57
<b><i>В.С. Привезенов</i></b> Техника обучения нейронной сети по распознаванию объекта на фото с использованием 3D-модели .....	61
<b><i>В.В. Разин, А.А. Краснов</i></b> Применение глубокого обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL .....	66
<b><i>В.В. Разин, А.А. Краснов</i></b> Применение машинного обучения в задачах диагностики заболевания по ЭКГ на наборе данных РТВ-XL .....	73
<b><i>М.И. Самбурова</i></b> Аугментация данных в обработке медицинских сигналов .....	80

<b><i>В.Н. Сорокина, М.М. Сорокина</i></b> Исследование классов групп, замкнутых относительно операций .....	92
<b><i>Е.Б. Титова</i></b> Об исследовании миноров матрицы инцидентности $d$ – мерного куба.....	99
<b><i>Д.А. Торбеева</i></b> Применение метода Монте-Карло для вычисления интегралов.....	103
<b><i>И.Х. Хуснуллин, З.И. Мухаметрахимова</i></b> Асимптотика собственного значения оператора Шредингера с растущим потенциалом .....	109
<b><i>К.А. Числова, Е.В. Борисова</i></b> Аппроксимация картографических объектов территорий, пострадавших от стихийных бедствий .....	112

# **Математика и ИТ – вместе в цифровое будущее**

Сборник трудов II Молодежной школы

*Нижегород, 24–28 апреля 2023 г.*

*Под ред. К.А. Баркалова, М.А. Богомоловой*

*Отв. за выпуск Т.Г. Смирнова*

Публикуется в авторской редакции

ННГУ им. Н.И. Лобачевского.  
603950, Н. Новгород, пр. Гагарина, 23.