

# ВЕСТНИК НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Научный журнал  
Основан в ноябре 1999 года

Серия: Информационные технологии

2025. Том 23, № 1

---

---

## СОДЕРЖАНИЕ

|                                                                                                                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Алеева В. Н.</i> Разработка программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов на основе концепции $Q$ -детерминанта..... | 5  |
| <i>Власов А. А., Романов Н. С.</i> Разработка программного модуля быстрого прослеживания границ в сейсмическом кубе с использованием алгоритма волновой трассировки .....                             | 19 |
| <i>Лулу Й. Г.</i> Сравнительный анализ статистических тестов на основе теста сжатия данных и стандартных тестов для оценки случайности генераторов случайных чисел .....                              | 33 |
| <i>Попов С. Е., Потапов В. П., Замараев Р. Ю.</i> Об одном подходе к разработке информационно-справочных систем на основе больших языковых моделей .....                                              | 46 |
| <i>Чёрный С. Г., Тарраф Д.</i> Об особенностях моделирования распада струи жидкости на капли в потоке газа .....                                                                                      | 67 |
| Информация для авторов .....                                                                                                                                                                          | 88 |



# V E S T N I K

## NOVOSIBIRSK STATE UNIVERSITY

Scientific Journal  
Since 1999, November  
In Russian

Series: Information Technologies

2025. Volume 23, № 1

---

---

### CONTENTS

|                                                                                                                                                                                                            |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Aleeva V. N.</i> Development of Software Systems for Computer-aided Design and Program Execution for the Effective Implementation of Numerical Algorithms Based on the Concept of $Q$ -determinant..... | 5  |
| <i>Vlasov A. A., Romanov N. S.</i> Development of a Software Module for Fast Tracing of Horizons in a Seismic Cube Ua Lee Algorithm.....                                                                   | 19 |
| <i>Lulu Yeshewas Getachew.</i> Comparative Analysis of Statistical Test based on Data compression methods and standard tests for assessing Randomness of Random Number Generators .....                    | 33 |
| <i>Popov S. E., Potapov V. P., Zamaraev R. Y.</i> On an Approach to Developing Information and Reference Systems Based on Large Language Models.....                                                       | 46 |
| <i>Cherny S. G., Tarraf D.</i> On the Features of Modeling the Primary Breakup of a Liquid Jet into Droplets in a Gas Flow .....                                                                           | 67 |
| Instructions for Contributors.....                                                                                                                                                                         | 88 |

*Editor in Chief* M. M. Lavrentiev

*Vice-Editor* A. V. Avdeev

*Executive Secretary* D. P. Iksanova

*Editorial Board of the Series*

- I. V. Bychkov*, professor, academician (Irkutsk), *B. M. Glinsky*, professor (Novosibirsk)  
*A. N. Gorban*, professor (Lester, GB), *E. P. Gordov*, professor (Tomsk)  
*B. S. Dobronets*, professor (Krasnoyarsk), *A. M. Elizarov*, professor (Kazan)  
*G. N. Erokhin*, professor (Kaliningrad), *A. I. Kamyshnikov*, professor (Khanty-Mansijsk)  
*G. P. Karev*, professor (Maryland, USA), *N. A. Kolchanov*, professor, academician (Novosibirsk)  
*M. M. Lavrentjev*, professor (Novosibirsk), *V. E. Malyshkin*, professor (Novosibirsk)  
*N. N. Mirenikov*, professor (Aizu, Japan), *N. M. Oskorbin*, professor (Barnaul)  
*D. E. Palchunov*, professor (Novosibirsk), *T. Pizansky*, professor (Ljubljana, Slovenia)  
*V. P. Potapov*, professor (Kemerovo), *O. I. Potaturkin*, professor (Novosibirsk)  
*V. A. Serebryakov*, professor (Moscow), *A. V. Starchenko*, professor (Tomsk)  
*S. I. Smagin*, professor, corresponding member of RAS (Khabarovsk)  
*D. A. Tusupov*, professor (Astana, Kazakhstan)  
*V. V. Shajdurov*, professor, corresponding member of RAS (Krasnoyarsk)  
*Yu. I. Shokin*, professor, academician (Novosibirsk)

*The journal is published quarterly in Russian since 1999  
by Novosibirsk State University Press*

*The address for correspondence  
Faculty of Information Technologies, Novosibirsk State University  
1 Pirogov Street, Novosibirsk, 630090, Russia  
Tel. +7 (383) 363 42 46*

*E-mail address:* [inftech@vestnik.nsu.ru](mailto:inftech@vestnik.nsu.ru)

*On-line version:* <http://elibrary.ru>

Научная статья

УДК 004.021, 004.032.24, 004.051, 004.272

DOI 10.25205/1818-7900-2025-23-1-5-18

## **Разработка программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов на основе концепции $Q$ -детерминанта**

**Валентина Николаевна Алеева**

Южно-Уральский государственный университет (НИУ)  
Челябинск, Россия

allevavn@susu.ru, <https://orcid.org/0000-0002-4717-0045>

### *Аннотация*

Статья посвящена решению актуальной проблемы разработки программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов. Теоретической основой описанного исследования является концепция  $Q$ -детерминанта, которая предлагает унифицированное представление численных алгоритмов в форме  $Q$ -детерминантов.  $Q$ -детерминант дает возможность выразить и оценить ресурс параллелизма алгоритма, а также показать способ его параллельного исполнения. В статье рассматриваются общие принципы организации программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов. Реализация этих принципов продемонстрирована на примерах создания конкретных программных систем. Кроме того, проведено сравнение подходов к реализации разработанных программных систем.

### *Ключевые слова*

$Q$ -детерминант алгоритма,  $Q$ -эффективная реализация алгоритма, ресурс параллелизма алгоритма,  $Q$ -эффективная программа, автоматизированное проектирование эффективных программ

### *Благодарности*

Как выпускник математического (теперь механико-математического) факультета Новосибирского государственного университета, выражаю благодарность факультету за данные мне знания.

### *Для цитирования*

Алеева В. Н. Разработка программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов на основе концепции  $Q$ -детерминанта // Вестник НГУ. Серия: Информационные технологии. 2025. Т. 23, № 1. С. 5–18. DOI 10.25205/1818-7900-2025-23-1-5-18

# Development of Software Systems for Computer-aided Design and Program Execution for the Effective Implementation of Numerical Algorithms Based on the Concept of $Q$ -determinant

Valentina N. Aleeva

South Ural State University  
Chelyabinsk, Russian Federation

allevavn@susu.ru, <https://orcid.org/0000-0002-4717-0045>

## Abstract

The paper is devoted to solving the actual problem of developing software systems for computer-aided design and program execution for the effective implementation of numerical algorithms. The theoretical basis of our research is the concept of  $Q$ -determinant. This concept suggests a unified representation of numerical algorithms in the form of  $Q$ -determinants. The  $Q$ -determinant makes it possible to express and evaluate the parallelism resource of the algorithm, as well as show the method of its parallel execution. In the paper we consider the general principles of software systems organizing for computer-aided design and program execution for the effective implementation of numerical algorithms. We demonstrate the realization of these principles by the examples of the creation of specific software systems. In addition, we compare the approaches to the implementation of the considered software systems.

## Keywords

$Q$ -determinant of algorithm,  $Q$ -effective implementation of algorithm, parallelism resource of algorithm,  $Q$ -effective program, computer-aided design of effective programs

## Acknowledgements

As I a graduate of the mathematical (now mechanics and mathematics) faculty of Novosibirsk State University, I would like to express my gratitude to the faculty for the knowledge given to me.

## For citation

Aleeva V. N. Development of software systems for computer-aided design and program execution for the effective implementation of numerical algorithms based on the concept of  $Q$ -determinant. *Vestnik NSU. Series: Information Technologies*, 2025, vol. 23, no. 1, pp. 5–18 (in Russ.) DOI 10.25205/1818-7900-2025-23-1-5-18

## Введение

Одним из подходов к распараллеливанию численных алгоритмов является концепция  $Q$ -детерминанта. Концепция и ее применение для решения проблемы эффективной реализации численных алгоритмов, численных методов и алгоритмических проблем на параллельных вычислительных системах (ПВС) описаны в работах [1–9]. Эффективные программы, разработанные с помощью концепции  $Q$ -детерминанта, называются  $Q$ -эффективными. Они выполняют  $Q$ -эффективные реализации алгоритмов, полностью использующие ресурсы параллелизма алгоритмов.

Для создания  $Q$ -эффективных программ был специально разработан метод проектирования [4]. Его может применять любой разработчик при проектировании для любого численного алгоритма программы, использующей ресурс параллелизма алгоритма полностью. Уметь создавать эффективные программы очень важно, но не менее важно иметь возможность автоматизированного проектирования эффективных программ.

Цель данного исследования заключается в том, чтобы показать применение концепции  $Q$ -детерминанта для создания единых для численных алгоритмов программных систем автоматизированного проектирования и исполнения  $Q$ -эффективных программ, при реализации которых могут использоваться различные подходы. Для достижения цели решаются следующие задачи.

1. Разработка общих принципов организации программных систем.
2. Сравнение подходов к реализации разработанных программных систем.

## Обзор работ по теме исследования

Одним из наиболее развитых направлений исследований в области параллельных вычислений является направление, созданное В. В. Воеводиным. Это направление представляет известная научная школа Вл. В. Воеводина. Именно оно наиболее близко к направлению, представленному в данной статье. Из работ, связанных с направлением, созданным В. В. Воеводиным, отметим работу [10], где проводится очень важное и развитое исследование параллельной структуры алгоритмов и программ для их реализации на ПВС. Исследование параллельной структуры алгоритмов основано на описании и изучении информационных графов алгоритмов. Оно применяется в открытой энциклопедии AlgoWiki [11]. Этот проект сформировался под руководством Вл. В. Воеводина и Дж. Донгарры (Jack Dongarra). В энциклопедии описываются и изучаются графы конкретных алгоритмов. Это позволяет исследовать ресурсы параллелизма алгоритмов и реализовывать их эффективно. Однако в работах данного направления исследований программное обеспечение для автоматизированного исследования и использования ресурсов параллелизма алгоритмов не рассматривается.

В сфере высокопроизводительных вычислений одним из ведущих ученых мира является Джек Донгарра. В университете Теннесси для поддержки библиотеки матрично-векторных операций DPLASMA [12] научным коллективом с участием Дж. Донгарры развивается подход к конструированию и исполнению параллельных программ на основе машинно-независимого представления прикладного алгоритма в виде бесконтурного ориентированного графа. Это позволяет обеспечивать высокопроизводительную реализацию прикладных алгоритмов благодаря планированию вычислений и динамической поддержке исполнительной системы PaRSEC [13]. Дж. Донгарра занимается также вопросами автоматического распараллеливания алгоритмов [14].

Еще один подход к эффективной реализации алгоритмов на ПВС разрабатывается в Стэнфордском университете под руководством А. Айкена (A. Aiken). Подход заключается в том, что отдельно описываются прикладной алгоритм и способ его отображения на вычислительные ресурсы ПВС. Научным коллективом, разрабатывающим подход, создана система Legion [15], реализующая этот подход. Недостатком системы является отсутствие автоматизации в конструировании способа отображения прикладного алгоритма на вычислительные ресурсы.

Существуют и другие подходы по теме исследования, однако на практике ресурс параллелизма алгоритмов при реализации на ПВС часто используется не полностью. Можно сделать вывод, что в настоящее время нет широко применяемых решений для эффективной реализации численных алгоритмов, тем более в автоматизированном режиме. По мнению автора, перспективным для решения проблемы автоматизированного проектирования и исполнения эффективных программ является подход на основе унифицированного представления алгоритма, показывающего ресурс параллелизма в полной мере. Например, подход, использующий концепцию  $Q$ -детерминанта.

## Концепция $Q$ -детерминанта

Теоретической основой данного исследования является концепция  $Q$ -детерминанта. Наиболее полно она изложена в [6]. Приведем основные понятия, необходимые для понимания исследования.

Пусть  $\alpha$  – численный алгоритм для решения алгоритмической проблемы  $\vec{y} = F(N, B)$ , где  $N = \{n_1, \dots, n_k\}$  – множество параметров размерности проблемы,  $k \geq 1$ , а  $n_i$  равно любому натуральному числу для каждого  $i \in \{1, \dots, k\}$ , либо  $N = \emptyset$ ,  $B$  – множество входных данных,  $\vec{y} = (y_1, \dots, y_m)$  – множество выходных данных,  $y_i \notin B$  для каждого  $i \in \{1, \dots, m\}$ . Целое число  $m$  является либо константой, либо значением вычислимой параметрической функции от  $N$  при условии, что  $N \neq \emptyset$ .

Пусть множество  $Q$  состоит из унарных и бинарных операций. Предположим, что алгоритм использует операции из множества  $Q$ . Вводятся понятия выражений над  $B$  и  $Q$ , их уровней вложенности и уровней вложенности операций. Выражения являются термами в стандартном смысле математической логики для обычной сигнатуры поля действительных чисел с пополнением операциями из множества  $Q$ .

Если  $N = \{n_1, \dots, n_k\}$ , то через  $\bar{N} = \{\bar{n}_1, \dots, \bar{n}_k\}$  обозначим кортеж, где  $\bar{n}_i$  – некоторое заданное значение параметра  $n_i$  для каждого  $i \in \{1, \dots, k\}$ , а через  $\{\bar{N}\}$  множество всех кортежей  $\bar{N}$ . Определим понятия  $Q$ -термов и их значений.

**Определение 1.** Если  $N = \emptyset$ , то любое выражение  $w$  над  $B$  и  $Q$  называется *безусловным  $Q$ -термом*. Пусть  $N \neq \emptyset$  и  $V$  – множество всех выражений над  $B$  и  $Q$ . Предположим, что есть отображение  $w: \bar{N} \rightarrow V \cup \emptyset$ . Тогда отображение называется *безусловным  $Q$ -термом*.

Таким образом, для  $N = \emptyset$  понятие безусловного  $Q$ -терма и выражения над  $B$  и  $Q$  совпадают. Если  $N \neq \emptyset$ , то для каждого  $\bar{N} \in \{\bar{N}\}$   $w(\bar{N})$  либо некоторое выражение над  $B$  и  $Q$ , либо  $w(\bar{N}) = \emptyset$ , что значит, что  $w(\bar{N})$  не определено.

**Определение 2.** Если  $N = \emptyset$ , то нахождение значения безусловного  $Q$ -терма  $w$  при любой интерпретации переменных  $B$  равносильно нахождению значения выражения  $w$ . Если  $N \neq \emptyset$  и  $w(\bar{N}) \neq \emptyset$ , то  $w(\bar{N})$  является выражением над  $B$  и  $Q$ . Мы можем найти значение выражения  $w(\bar{N})$ . Конечно, мы опускаем значение  $w(\bar{N}) = \emptyset$ . Следовательно, нахождение значения безусловного  $Q$ -терма  $w$  при любой интерпретации переменных  $B$  определено.

**Определение 3.** Пусть  $N = \emptyset$  и  $w$  – безусловный  $Q$ -терм. Предположим, что выражение  $w$  над  $B$  и  $Q$  имеет значение логического типа при любой интерпретации переменных  $B$ . В этом случае  $Q$ -терм называется *безусловным логическим  $Q$ -термом*. Пусть  $N \neq \emptyset$  и  $w$  – безусловный  $Q$ -терм. Если выражение  $w(\bar{N})$  для любого  $\bar{N} \in \{\bar{N}\}$  имеет значение логического типа при любой интерпретации переменных  $B$ , то безусловный  $Q$ -терм  $w$  называется *безусловным логическим  $Q$ -термом*.

**Определение 4.** Пусть  $u_1, \dots, u_l$  – безусловные логические  $Q$ -термы,  $w_1, \dots, w_l$  – безусловные  $Q$ -термы. Введем обозначение  $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$  и будем называть его *условным  $Q$ -термом длины  $l$* .

Опишем нахождение значения условного  $Q$ -терма  $(\hat{u}, \hat{w})$  при интерпретации переменных  $B$ .

**Определение 5.** Пусть  $N = \emptyset$ . Находим значения выражений  $u_i, w_i$  для  $i \in \{1, \dots, l\}$ . При таком нахождении значений можно найти пару  $u_{i_0}, w_{i_0}$  такую, что  $u_{i_0}$  имеет значение *true*. Следовательно, можно найти значение  $w_{i_0}$ . После этого считаем, что  $(\hat{u}, \hat{w})$  имеет значение  $w_{i_0}$ . В противном случае считаем, что значение  $(\hat{u}, \hat{w})$  при данной интерпретации переменных  $B$  не определено. В случае  $N \neq \emptyset$  и  $\bar{N} \in \{\bar{N}\}$  нахождение значения условного  $Q$ -терма выполняется аналогично.

**Определение 6.** Пусть  $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, 2, \dots\}}$  – счетное множество пар безусловных  $Q$ -термов. Предположим, что  $(\hat{u}, \hat{w}) = \{(u_i, w_i)\}_{i \in \{1, \dots, l\}}$  является условным  $Q$ -термом длины  $l$  для любого  $l < \infty$ . Тогда мы называем  $(\hat{u}, \hat{w})$  *условным бесконечным  $Q$ -термом*.

Опишем нахождение значения условного бесконечного  $Q$ -терма  $(\hat{u}, \hat{w})$  при интерпретации переменных  $B$ .

**Определение 7.** Пусть  $N = \emptyset$ . Находим значения выражений  $u_i, w_i$  для  $i \in \{1, 2, \dots\}$ . В результате можно найти пару  $u_{i_0}, w_{i_0}$  такую, что  $u_{i_0}$  имеет значение *true*. Следовательно, можно найти значение  $w_{i_0}$ . После этого считаем, что  $(\hat{u}, \hat{w})$  имеет значение  $w_{i_0}$ . В противном случае считаем, что значение  $(\hat{u}, \hat{w})$  при данной интерпретации переменных  $B$  не определено. В случае  $N \neq \emptyset$  и  $\bar{N} \in \{\bar{N}\}$  нахождение значения условного бесконечного  $Q$ -терма выполняется аналогично.

**Замечание 1.** Если не имеет значения, является ли  $Q$ -терм безусловным, условным или условным бесконечным, то мы называем его  $Q$ -термом.

**Определение 8.** Предположим, что алгоритм  $\alpha$  состоит в том, что для каждого  $i \in M$  где  $M = \{1, \dots, m\}$ , нужно найти  $y_i$ , как значение  $Q$ -терма  $f_i$ . Тогда множество  $Q$ -термов  $\{f_i \mid i \in M\}$  называется  $Q$ -детерминантом алгоритма  $\alpha$ , а система уравнений  $y_i = f_i$  для всех  $i \in M$  представлением алгоритма  $\alpha$  в форме  $Q$ -детерминанта.

Пусть  $U$ ,  $C$  и  $I$  образуют разбиение множества  $M = \{1, \dots, m\}$ , удовлетворяющее условиям:

1.  $U \cup C \cup I = M$ ;
2.  $U \cap C = U \cap I = C \cap I = \emptyset$ ;
3. Одно или два подмножества  $U$ ,  $C$  и  $I$  могут быть пустыми.

При этом  $U$ ,  $C$  и  $I$  связаны с подмножествами множества  $Q$ -термов  $\{f_i\}_{i \in M}$  следующим образом:

1. Для каждого  $i \in U$  есть  $Q$ -терм  $f_i$ , который является безусловным, и  $f_i = w^i$ ;
2. Для каждого  $i \in C$  есть  $Q$ -терм  $f_i$ , который является условным, и  $f_i = \left\{ (u_j^i, w_j^i) \right\}_{j \in \{1, \dots, l(i)\}}$ ,

где  $l(i)$  – либо константа, либо значение вычислимой функции от  $N$ , если  $N \neq \emptyset$ ;

3. Для каждого  $i \in I$  есть  $Q$ -терм  $f_i$ , являющийся условным бесконечным, и  $f_i = \left\{ (u_j^i, w_j^i) \right\}_{j \in \{1, 2, \dots\}}$ .

**Определение 9.** Пусть алгоритм  $\alpha$  представлен в форме  $Q$ -детерминанта  $y_i = f_i$  для всех  $i \in M$ . Процесс вычисления  $Q$ -термов  $f_i$  для всех  $i \in M$  называется *реализацией алгоритма  $\alpha$* . Если реализация алгоритма  $\alpha$  такова, что две или более операций выполняются одновременно, то она называется *параллельной реализацией*.

**Определение 10.** Реализация алгоритма называется  $Q$ -эффективной, если выражения

$$W(\bar{N}) = \left\{ w^i(\bar{N}) (i \in U); u_j^i(\bar{N}), w_j^i(\bar{N}) (i \in C, j \in \{1, \dots, l(i)\}); u_j^i(\bar{N}), w_j^i(\bar{N}) \right. \\ \left. (i \in I, j \in \{1, 2, \dots\}) \right\}$$

вычисляются одновременно, при этом их операции выполняются по мере вычисления операндов.

При вычислении выражений выполняются следующие условия. Если для любых  $i \in C \cup I$  и  $i \in \{1, 2, \dots\}$  выражение  $u_j^i(\bar{N})$  имеет значение *false*, то вычисление соответствующего выражения  $w_j^i(\bar{N})$  завершается. Если для любых  $i \in C \cup I$  и  $i \in \{1, 2, \dots\}$  вычисление некоторой пары выражений  $(u_j^i(\bar{N}), w_j^i(\bar{N}))$  приводит к тому, что значение одного из двух выражений не определено, то вычисление другого выражения прекращается. Если для любого  $i \in C \cup I$  вычисление некоторой пары выражений  $(u_{j_0}^i(\bar{N}), w_{j_0}^i(\bar{N}))$  приводит к нахождению их значений и  $u_{j_0}^i(\bar{N})$  равно *true*, то вычисление выражений  $(u_j^i(\bar{N}), w_j^i(\bar{N}))$  прекращается для любого  $j \neq j_0$ . Если  $N \neq \emptyset$ , то  $\bar{N}$  в выражениях  $W(\bar{N})$  не используется.

**Замечание 2.** Определение  $Q$ -эффективной реализации показывает, что это *наиболее параллельная реализация алгоритма*. Другими словами,  $Q$ -эффективная реализация полностью использует ресурс параллелизма алгоритма.

## Общие принципы организации программных систем

Программные системы автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов, разработанные на основе концепции  $Q$ -детерминанта, должны включать две подсистемы, решающие следующие задачи.



**Пример 2.** Предположим, что матрица  $A$  размера  $2 \times 2$  обратима (имеет ненулевой определитель). Пусть  $\vec{X} = (X(1), X(2))^T$  и  $\vec{B} = (A(1,3), A(2,3))^T$  – векторы-столбцы,  $\bar{A} = [A(i, j)]_{2 \times 3}$  – расширенная матрица системы  $A\vec{X} = \vec{B}$ . Рис. 2 содержит представление в форме  $Q$ -детерминанта алгоритма решения системы линейных уравнений 2-го порядка методом Гаусса – Жордана, использующего в качестве ведущего элемента первый ненулевой элемент ведущей строки.  $Q$ -детерминант алгоритма состоит из двух условных  $Q$ -термов длины 2.

```

X(1)={"op": "!=" , "f0": "A(1,1)", "s0": "0"};
      {"op": "-", "f0": {"op": "/", "f0": "A(1,3)",
                        "s0": "A(1,1)"},
      "s0": {"op": "*", "f0": {"op": "/", "f0": {"op": "-", "f0": "A(2,3)",
                                                "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,3)",
                                                                "s0": "A(1,1)"},
                                                "s0": "A(2,1)"},
                        "s0": {"op": "-", "f0": "A(2,2)",
                        "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,2)",
                                                                "s0": "A(1,1)"},
                        "s0": "A(2,1)"}}}
      "s0": {"op": "/", "f0": "A(1,2)",
      "s0": "A(1,1)"}}}
X(2)={"op": "!=" , "f0": "A(1,1)", "s0": "0"};
      {"op": "/", "f0": {"op": "-", "f0": "A(2,3)",
                        "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,3)",
                                                                "s0": "A(1,1)"},
                        "s0": "A(2,1)"},
      "s0": {"op": "-", "f0": "A(2,2)",
      "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,2)",
                                                                "s0": "A(1,1)"},
      "s0": "A(2,1)"}}}
X(1)={"op": "=", "f0": "A(1,1)", "s0": "0"};
      {"op": "/", "f0": {"op": "-", "f0": "A(2,3)",
                        "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,3)",
                                                                "s0": "A(1,2)"},
                        "s0": "A(2,2)"},
      "s0": {"op": "-", "f0": "A(2,1)",
      "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,1)",
                                                                "s0": "A(1,2)"},
      "s0": "A(2,2)"}}}
X(2)={"op": "=", "f0": "A(1,1)", "s0": "0"};
      {"op": "-", "f0": {"op": "/", "f0": "A(1,3)",
                        "s0": "A(1,2)"},
      "s0": {"op": "*", "f0": {"op": "/", "f0": {"op": "-", "f0": "A(2,3)",
                                                                "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,3)",
                                                                "s0": "A(1,2)"},
                                                                "s0": "A(2,2)"},
                        "s0": {"op": "-", "f0": "A(2,1)",
                        "s0": {"op": "*", "f0": {"op": "/", "f0": "A(1,1)",
                                                                "s0": "A(1,2)"},
                        "s0": "A(2,2)"}}}
      "s0": {"op": "/", "f0": "A(1,1)",
      "s0": "A(1,2)"}}}

```

Рис. 2. Представление в форме  $Q$ -детерминанта алгоритма решения системы линейных уравнений второго порядка методом Гаусса – Жордана

Fig. 2. Representation in the form of a  $Q$ -determinant of the algorithm for solving a system of linear equations of the second order using the Gauss – Jordan method

На рис. 1 и рис. 2 строки файлов, содержащих представления алгоритмов в форме  $Q$ -детерминантов, размещены так, чтобы показать уровни вложенности операций.

## Сравнение подходов к реализации разработанных программных систем

Рассмотрим следующие программные системы автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов, разработанные в настоящее время:

- 1) на основе компилятора и виртуальной машины;
- 2) лексического анализатора.

В их разработку внесли вклад студенты кафедры системного программирования Южно-Уральского государственного университета А. В. Юферов и П. А. Манатин.

Программная система на основе компилятора и виртуальной машины описана в работе [16], а на основе лексического анализатора в работе [17], кроме того, о ней сделан доклад на научной конференции «Параллельные вычислительные технологии (ПаВТ) 2024». Проанализируем и сравним основные решения по реализации систем.

В состав программной системы на основе компилятора и виртуальной машины входят:

- 1) компилятор – преобразователь представления алгоритма в форме  $Q$ -детерминанта в код программы для выполнения  $Q$ -эффективной реализации алгоритма;
- 2) виртуальная машина – исполнитель создаваемых компилятором программ.

Аналогично в состав программной системы на основе лексического анализатора входят:

- 1) программа для формирования представления  $Q$ -эффективной реализации численного алгоритма на базе лексического анализатора – далее лексический анализатор;
- 2) программа для выполнения представления  $Q$ -эффективной реализации численного алгоритма.

Таким образом, обе программные системы содержат две необходимые для таких систем подсистемы. При этом они применяют представления алгоритмов в форме  $Q$ -детерминантов, полученные с помощью программной  $Q$ -системы. Следовательно, разработанные программные системы используют общие принципы организации программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов, описанные выше. Далее кратко опишем реализацию каждой из разработанных программных систем.

Диаграмма размещения компонентов программной системы на основе компилятора и виртуальной машины представлена на рис. 3.

Поясним диаграмму. Подкомпонент «Обработчик  $Q$ -детерминанта алгоритма» компилятора, используя  $Q$ -детерминант алгоритма, строит в памяти дерево алгоритма – структуру, содержащую операции  $Q$ -детерминанта и связи между ними, а затем подкомпонент «Генератор программ» компилятора записывает в бинарном виде метаданные алгоритма и полученную с помощью дерева алгоритма очередь команд для выполнения  $Q$ -эффективной реализации алгоритма в файл программы. Файл метаданных алгоритма – текстовый файл, создаваемый пользователем, который используется компилятором для информации о входных данных алгоритма, их описания, а также текстового описания самого алгоритма.

Виртуальная машина загружает выбранную пользователем программу в память, читает входные данные алгоритма из файла, содержащего входные данные, получает поток команд из программы. Подкомпонент «Диспетчер ВМ» виртуальной машины обеспечивает распределение команд по потокам исполнения, в которых работают объекты компонента «Интерпретатор байт-кода». Команды выполняются по мере их готовности к выполнению. После выполнения всех команд программы виртуальная машина записывает полученные выходные данные в файл для выходных данных. При исполнении программы оперативная память используется непосредственно, что обеспечивает высокое быстродействие. Для разработки программного обеспечения компилятора и виртуальной машины применялось объектно-ориентированное программирование.

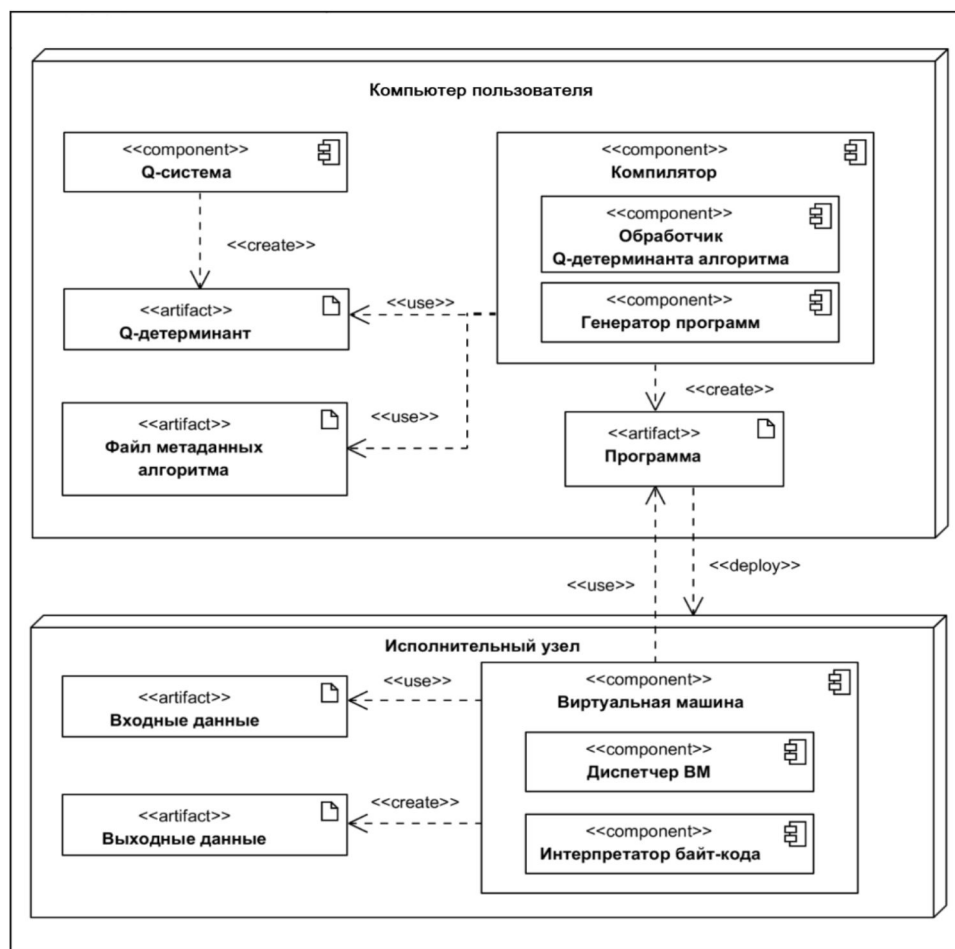


Рис. 3. Диаграмма размещения компонентов программной системы на основе компилятора и виртуальной машины

Fig. 3. Deployment diagram of the software system based on a compiler and a virtual machine

Программная система на основе компилятора и виртуальной машины при разработке была ориентирована на ПВС с общей памятью. Затем было разработано решение по ее модификации с целью возможности эксплуатации на ПВС с распределенной памятью. Это решение использует принцип MASTER-SLAVE [18], а также технологии OpenMP и MPI.

Узел MASTER распределяет задачи по узлам SLAVE и агрегирует результаты вычислений, в то время как подчиненные узлы SLAVE непосредственно производят вычисления. Технология MPI применяется для распределения задач по вычислительным узлам. Для распределения вычислений по потокам на одном узле используется технология OpenMP.

Узел MASTER принимает на вход двоичный файл программы, созданной компилятором, и входные данные. Если  $Q$ -детерминант алгоритма состоит из безусловных  $Q$ -термов, то узел MASTER определяет, какие из них могут быть вычислены, и передает их узлам SLAVE. Оставшиеся безусловные  $Q$ -термы передаются на исполнение по мере готовности. Если  $Q$ -детерминант алгоритма состоит из условных  $Q$ -термов, то узел MASTER определяет, какие пары условных  $Q$ -термов, состоящие из безусловного логического и безусловного  $Q$ -термов, могут быть вычислены, и передает их узлам SLAVE. Оставшиеся пары  $Q$ -термов передаются на исполнение по мере готовности.  $Q$ -термы из пары вычисляются параллельно, при этом, если логический  $Q$ -терм принимает значение *false*, вычисление второго  $Q$ -терма из пары прекращается, а на узел MASTER передается соответствующее сообщение.

Рис. 4 содержит диаграмму размещения компонентов программной системы на основе лексического анализатора. Дадим пояснения к диаграмме. Лексический анализатор выполняет считывание описания представления алгоритма в форме  $Q$ -детерминанта из файла формата JSON, полученного с помощью  $Q$ -системы, формирование представления  $Q$ -эффективной реализации алгоритма, вывод полученных результатов в виде директорий с файлами по определенным описанным далее правилам. В этих файлах описана булева функция, приведенная к конъюнктивной форме записи.

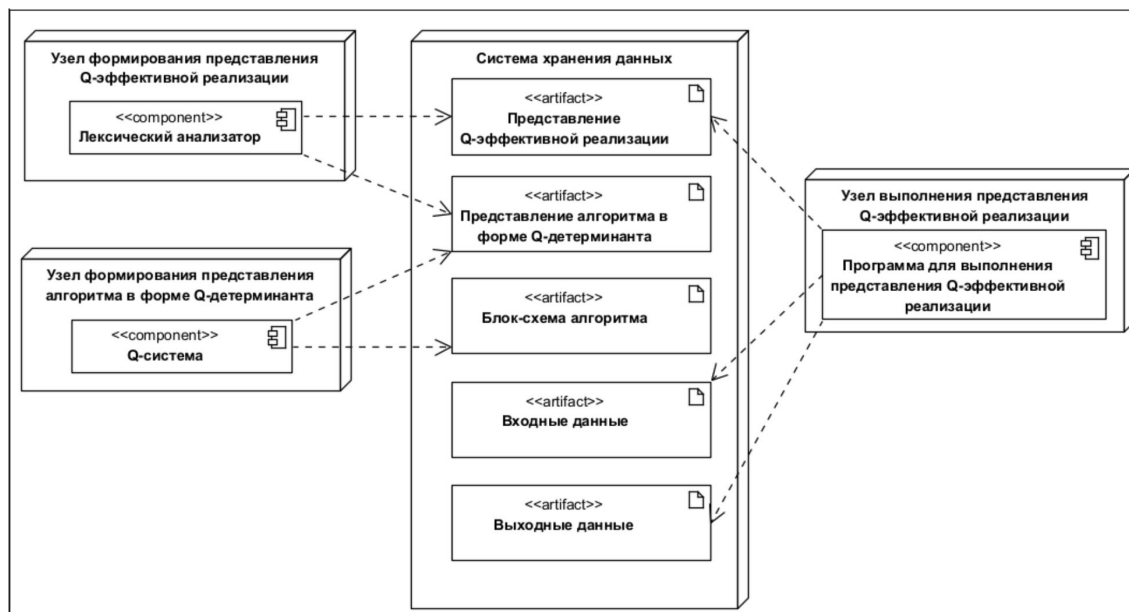


Рис. 4. Диаграмма размещения компонентов программной системы на основе лексического анализатора

Fig. 4. Deployment diagram of the software system based on a lexical analyzer

Лексический анализатор спроектирован в соответствии с парадигмой объектно-ориентированного программирования. Были созданы классы  $QString$  и  $QTerm$ . Принцип работы лексического анализатора заключается в следующем. В конструкторе класса  $QString$  исходная строка представления алгоритма в форме  $Q$ -детерминанта разбивается на две части: строку, содержащую логический  $Q$ -терм (при его наличии), и строку, содержащую безусловный  $Q$ -терм. Эти строки передаются конструктору класса  $QTerm$  для лексического анализа. Анализ происходит до тех пор, пока вложенные объекты  $QTerm$  не будут представлять собой выражения, содержащие одну операцию сравнения или одну арифметическую операцию. Каждый такой объект по окончании инициализации себя и вложенных объектов добавляет своему предку в словарь операций и множество данных значения своих соответствующих полей. Таким образом, корневой  $QTerm$  содержит результат лексического анализа соответствующего объекта  $QString$ .

После лексического анализа строк лексический анализатор создает представление  $Q$ -эффективной реализации, группируя полученные объекты  $QString$  по некоторым правилам, чтобы иметь возможность разделить исходную задачу на несколько подзадач, которые можно выполнять на системах с распределенной памятью, а также использовать для распараллеливания задачи при выполнении на системах с общей памятью.

Представление  $Q$ -эффективной реализации алгоритма состоит из директорий. Каждая директория содержит файлы, описывающие вычисление одного из элементов выходных данных. Таким образом, количество директорий соответствует количеству элементов выходных данных. Действия над директориями могут проводиться параллельно на разных вычисли-

тельных устройствах. В таком случае можно обеспечить локальность данных при выполнении алгоритма.

Если выходная переменная вычисляется с помощью безусловного  $Q$ -терма, то соответствующая ей директория содержит один файл с одной строкой, которая представляет собой этот безусловный  $Q$ -терм. Если выходная переменная вычисляется с помощью условного  $Q$ -терма, то соответствующая ей директория содержит количество файлов, равное длине условного  $Q$ -терма. В этом случае каждый файл соответствует одной паре безусловных  $Q$ -термов условного  $Q$ -терма. Первая строка файла представляет собой второй  $Q$ -терм пары, а остальные строки – выражения с одной операцией сравнения, полученные лексическим анализатором из логического  $Q$ -терма пары. Строки каждого из этих файлов можно вычислять параллельно.

Программа для выполнения представления  $Q$ -эффективной реализации выполняет  $Q$ -эффективную реализацию алгоритма, используя ее представление, сформированное лексическим анализатором. Перед выполнением программа читает входные данные алгоритма из файла специального формата, содержащего входные данные, результат выполнения она записывает в файл для выходных данных.

Выполнение  $Q$ -эффективной реализации с помощью ее представления заключается в следующем. Если выходная переменная вычисляется с помощью безусловного  $Q$ -терма, то ее значение определяется путем вычисления безусловного  $Q$ -терма, содержащегося в файле, соответствующем выходной переменной. Если выходная переменная вычисляется с помощью условного  $Q$ -терма, то для определения ее значения вычисляются  $Q$ -термы и выражения, содержащиеся во всех строках файлов, соответствующих выходной переменной, и выходной переменной присваивается результат вычисления безусловного  $Q$ -терма файла, все остальные строки которого содержат выражения, имеющие значение *true*. Программа для выполнения представления  $Q$ -эффективной реализации может использоваться как для общей, так и для распределенной памяти ПВС.

Каждый из процессов построения представления  $Q$ -эффективной реализации и его выполнения может осуществляться на своей вычислительной системе.

Сравнивая решения, применяемые для реализации рассмотренных программных систем, приходим к выводу, что подходы к разработке систем существенно различаются, несмотря на то, что системы используют общие принципы организации программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов.

Следует отметить, что организация хранения данных алгоритма и представления  $Q$ -эффективной реализации алгоритма значительно влияет на быстродействие программной системы. Для организации хранения данных программной системы на основе компилятора и виртуальной машины используется оперативная память, а для организации хранения данных программной системы на основе лексического анализатора применяются файлы, поэтому быстродействие первой программной системы выше, чем второй. Это различие в быстродействии подтверждают экспериментальные исследования, приведенные в работах [16; 17].

Таким образом, при разных подходах к разработке программных систем быстродействие может быть разным. Кроме уже реализованных подходов, конечно, возможны и другие подходы.

## Заключение

Исследования, приведенные в данной статье, показывают, что разработка программных систем автоматизированного проектирования и исполнения программ для эффективной реализации численных алгоритмов на основе концепции  $Q$ -детерминанта возможна. При этом для реализации программных систем могут использоваться разные подходы, обеспечивающие разное быстродействие. Вместе с тем программные системы используют общие принципы организации.

Ценность программных систем автоматизированного проектирования и исполнения программ состоит в том, что нет необходимости в использовании труда высококвалифицированных программистов.

### Список литературы

1. **Алеева В. Н.** Анализ параллельных численных алгоритмов. Препринт № 590. Новосибирск: ВЦ СО АН СССР, 1985. 23 с.
2. **Алеева В. Н., Зотова П. С., Склезнев Д. С.** Расширение возможностей исследования ресурса параллелизма численных алгоритмов с помощью программной Q-системы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 2. С. 66–81. DOI: 10.14529/cmse210205
3. **Алеева В. Н., Шатов М. Б.** Применение концепции Q-детерминанта для эффективной реализации численных алгоритмов на примере метода сопряженных градиентов для решения систем линейных уравнений // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2021. Т. 10, № 3. С. 56–71. DOI: 10.14529/cmse210304
4. **Aleeva V.** Designing a Parallel Programs on the Base of the Conception of Q-Determinant // Supercomputing. RuSCDays 2018. Communications in Computer and Information Science. 2019. Vol. 965. P. 565–577. DOI: 10.1007/978-3-030-05807-4\_48
5. **Aleeva V. N.** Improving Parallel Computing Efficiency // Proceedings – 2020 Global Smart Industry Conference, GloSIC 2020. IEEE. 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828
6. **Valentina Aleeva, Rifkhat Aleev.** Investigation and Implementation of Parallelism Resources of Numerical Algorithms // ACM Transactions on Parallel Computing. 2023. Vol. 10. no. 2, Article number 8. P. 1–64. DOI: 10.1145/3583755
7. **Aleeva V. N., Aleev R. Zh.** High-Performance Computing Using Application of Q-determinant of Numerical Algorithms // Proceedings – 2018 Global Smart Industry Conference, GloSIC 2018. IEEE. 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160
8. **Aleeva V., Bogatyreva E., Skleznev A., et al.** Software Q-system for the Research of the Resource of Numerical Algorithms Parallelism // Supercomputing. RuSCDays 2019. Communications in Computer and Information Science. 2019. Vol. 1129. P. 641–652. DOI: 10.1007/978-3-030-36592-9\_52
9. **Aleeva V. N., Sharabura I. S., Suleymanov D. E.** Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q-determinant // Parallel Computing Technologies (PaCT 2015). Lecture Notes in Computer Science. 2015. Vol. 9251. P. 3–9. DOI: 10.1007/978-3-319-21909-7\_1
10. **Воеводин В. В., Воеводин Вл. В.** Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
11. **Antonov A. S., Dongarra J., Voevodin V. V.** AlgoWiki Project as an Extension of the Top500 Methodology // Supercomputing Frontiers and Innovations. 2018. Vol. 5, no. 1. P. 4–10. DOI: 10.14529/jsfi180101
12. **Bosilca G., Bouteiller A., Danalis A., et al.** Flexible Development of Dense Linear Algebra Algorithms on Massively Parallel Architectures with DPLASMA // 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum. 2011. P. 1432–1441. DOI: 10.1109/IPDPS.2011.299
13. **Bosilca G., Bouteiller A., Danalis A., et al.** PaRSEC: Exploiting Heterogeneity to Enhance Scalability // Computing in Science & Engineering. 2013. Vol. 15, no. 6. P. 36–45. DOI: 10.1109/MCSE.2013.98

14. **Balaprakash P., Dongarra J., Gamblin T., et al.** Autotuning in High-Performance Computing Applications // *Proceedings of the IEEE*. 2018. Vol. 106, no. 11. P. 2068–2083. DOI: 10.1109/JPROC.2018.2841200
15. **Bauer M., Treichler S., Slaughter E., and A. Aiken A.** Legion: Expressing locality and independence with logical regions // *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 2012. P. 1–11. DOI: 10.1109/SC.2012.71
16. **Алеева В. Н.** Автоматизированное проектирование и исполнение эффективных программ для численных алгоритмов // *Вестник ЮУрГУ. Серия: Вычислительная математика и информатика*. 2023. Т. 12, № 3. С. 31–49. DOI: 10.14529/cmse230303
17. **Manatin P., Aleeva V.** Efficient Implementation of Numerical Algorithms Based on a Lexical Analyzer // *Parallel Computational Technologies. PCT 2024. Communications in Computer and Information Science*. 2024. Vol 2241. P. 107–121. DOI: 10.1007/978-3-031-73372-7\_8
18. **Leung J. Y.-T., Zhao H.** Scheduling problems in master-slave model // *Annals of Operations Research*. 2008. Vol. 159. P. 215–231. DOI: 10.1007/s10479-007-0271-4

### References

1. **Aleeva V. N.** Analysis of Parallel Numerical Algorithms. Preprint no. 590. Novosibirsk, Computing Center of the Siberian Branch of the Academy of Sciences of the USSR, 1985. 23 p. (in Russian)
2. **Aleeva V. N., Zotova P. S., Skleznev D. S.** Advancement of Research for the Parallelism Resource of Numerical Algorithms with the Help of Software Q-system. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2021, vol. 10, no. 2, pp. 66–81. (in Russian) DOI: 10.14529/cmse210205
3. **Aleeva V. N., Shatov M. B.** Application of the Q-determinant Concept for Efficient Implementation of Numerical Algorithms by the Example of the Conjugate Gradient Method for Solving Systems of Linear Equations. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2021, vol. 10, no. 3, pp. 56–71. (in Russian) DOI: 10.14529/cmse210304
4. **Aleeva V.** Designing a Parallel Programs on the Base of the Conception of Q-Determinant. Supercomputing. RuSCDays 2018. *Communications in Computer and Information Science*, 2019, vol. 965, pp. 565–577. DOI: 10.1007/978-3-030-05807-4\_48
5. **Aleeva V. N.** Improving Parallel Computing Efficiency. Proceedings – 2020 Global Smart Industry Conference, GloSIC 2020. IEEE. 2020. P. 113–120. Article number 9267828. DOI: 10.1109/GloSIC50886.2020.9267828
6. **Aleeva V., Aleev R.** Investigation and Implementation of Parallelism Resources of Numerical Algorithms. *ACM Transactions on Parallel Computing*, 2023, vol. 10, no. 2, pp. 1–64. DOI: 10.1145/3583755
7. **Aleeva V. N., Aleev R. Zh.** High-Performance Computing Using Application of Q-determinant of Numerical Algorithms. Proceedings – 2018 Global Smart Industry Conference, GloSIC 2018. IEEE. 2018. 8 p. Article number 8570160. DOI: 10.1109/GloSIC.2018.8570160
8. **Aleeva V., Bogatyreva E., Skleznev A., et al.** Software Q-system for the Research of the Resource of Numerical Algorithms Parallelism. Supercomputing. RuSCDays 2019. *Communications in Computer and Information Science*, 2019, vol. 1129, pp. 641–652. DOI: 10.1007/978-3-030-36592-9\_52
9. **Aleeva V. N., Sharabura I. S., Suleymanov D. E.** Software System for Maximal Parallelization of Algorithms on the Base of the Conception of Q-determinant. *Parallel Computing Technologies*

- (PaCT 2015). *Lecture Notes in Computer Science*, 2015, vol. 9251, pp. 3–9. DOI: 10.1007/978-3-319-21909-7\_1
10. **Voevodin V. V., Voevodin V. V.** Parallel Computing. St.Petersburg, BHV-Petersburg, 2002. 608 p. (in Russian)
  11. **Antonov A. S., Dongarra J., Voevodin V. V.** AlgoWiki Project as an Extension of the Top500 Methodology. *Supercomputing Frontiers and Innovations*, 2018, vol. 5, no. 1, pp. 4–10. DOI: 10.14529/jsfi180101
  12. **Bosilca G., Bouteiller A., Danalis A., et al.** Flexible Development of Dense Linear Algebra Algorithms on Massively Parallel Architectures with DPLASMA. *2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum*, 2011. P. 14321441. DOI: 10.1109/IPDPS.2011.299
  13. **Bosilca G., Bouteiller A., Danalis A., et al.** PaRSEC: Exploiting Heterogeneity to Enhance Scalability. *Computing in Science & Engineering*, 2013, vol. 15, no. 6, pp. 36–45. DOI: 10.1109/MCSE.2013.98
  14. **Balaprakash P., Dongarra J., Gamblin T., et al.** Autotuning in High-Performance Computing Applications. *Proceedings of the IEEE*, 2018, vol. 106, no. 11, pp. 2068–2083. DOI: 10.1109/JPROC.2018.2841200
  15. **Bauer M., Treichler S., Slaughter E., and A. Aiken A.** Legion: Expressing locality and independence with logical regions. *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 2012. P. 1–11. DOI: 10.1109/SC.2012.71
  16. **Aleeva V. N.** Computer-aided design and execution of effective programs for numerical algorithms. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2023. Vol. 12, no. 3. P. 31–49. (in Russian) DOI: 10.14529/cmse230303
  17. **Manatin P., Aleeva V.** Efficient Implementation of Numerical Algorithms Based on a Lexical Analyzer. Parallel Computational Technologies. PCT 2024. *Communications in Computer and Information Science*, 2024, vol. 2241, pp. 107–121. DOI: 10.1007/978-3-031-73372-7\_8
  18. **Leung J. Y.-T., Zhao H.** Scheduling problems in master-slave mode. *Annals of Operations Research*, 2008, vol. 159, pp. 215–231. DOI: 10.1007/s10479-007-0271-4

### Информация об авторе

Алеева Валентина Николаевна, кандидат физико-математических наук

### Information about the Author

Valentina N. Aleeva, Candidate of Science in Physics and Mathematics

Статья поступила в редакцию 27.12.2024;  
одобрена после рецензирования 14.01.2025; принята к публикации 14.01.2024

The article was submitted 27.12.2024;  
approved after reviewing 14.01.2025; accepted for publication 14.01.2025

Научная статья

УДК 550.8.05

DOI 10.25205/1818-7900-2025-23-1-19-32

## **Разработка программного модуля быстрого прослеживания границ в сейсмическом кубе с использованием алгоритма волновой трассировки**

**Александр Александрович Власов<sup>1,2</sup>**  
**Никита Сергеевич Романов<sup>1</sup>**

<sup>1</sup>Новосибирский государственный университет  
Новосибирск, Россия

<sup>2</sup>Институт автоматизации и электрометрии СО РАН  
Новосибирск, Россия

a.vlasov7@g.nsu.ru  
n.romanov1@g.nsu.ru

### *Аннотация*

Для оценки месторождений полезных ископаемых и определения потенциальных зон для бурения по данным сейсморазведки необходимо построить согласованную сейсмостратиграфическую модель. Построение такой модели производится в несколько этапов: полевые работы, первичная обработка данных, расчет трансформированных сигналов и интерпретация. Согласованность результатов интерпретации достигается за счет соблюдения основных стратиграфических принципов. В работе рассмотрены два подхода автоматизации определения отражающих горизонтов в сейсмических кубах с учетом априорной информации на основе алгоритма волновой трассировки. Отличительной особенностью предложенного решения является: скорость работы, малая доля последовательного исполнения, устойчивость решения, учет ограничений, выставленных пользователем, соблюдение двух стратиграфических принципов (первичной горизонтальности и суперпозиции). В тестовом сейсмическом кубе размером 27951 трасс, с размерностью 200 измерений с двумя заданными пользователем границами, выполнено прослеживание горизонтов за 12 секунд с использованием двойного объема оперативной памяти от размера обрабатываемых данных. Разработанный подход не решает задачу выделения разломов, но способен учитывать их наличие за счет существующей возможности задания фиксированных границ, прослеженных пользователем в качестве аргумента функции.

### *Ключевые слова*

сейсморазведка, отражающие горизонты, стратиграфические границы, функция взаимной корреляции, автоматизация процесса, алгоритм волновой трассировки

### *Для цитирования*

Власов А. А., Романов Н. С. Разработка программного модуля быстрого прослеживания границ в сейсмическом кубе с использованием алгоритма волновой трассировки // Вестник НГУ. Серия: Информационные технологии. 2025. Т. 23, № 1. С. 19–32. DOI 10.25205/1818-7900-2025-23-1-19-32

## Development of a Software Module for Fast Tracing of Horizons in a Seismic Cube Using Lee Algorithm

Alexander A. Vlasov<sup>1,2</sup>, Nikita S. Romanov<sup>1</sup>

<sup>1</sup>Novosibirsk State University,  
Novosibirsk, Russian Federation

<sup>2</sup>Institute of Automation and Electrometry SB RAS,  
Novosibirsk, Russian Federation

a.vlasov7@g.nsu.ru  
n.romanov1@g.nsu.ru

### Abstract

For the assessment of mineral deposits and the identification of potential drilling zones using seismic data, it is necessary to build a consistent seismic stratigraphic model. The construction of such a model is carried out in several stages: field-work, initial data processing, calculation of transformed signals, and interpretation. Consistency in the interpretation results is achieved by adhering to fundamental stratigraphic principles. This work considers two approaches to automating the identification of reflection horizons in seismic cubes, considering prior information based on Lee algorithm. A distinctive feature of the proposed solution is its speed, low proportion of sequential execution, solution stability, consideration of constraints set by the user, and adherence to two stratigraphic principles (original horizontality and superposition). In a test seismic cube with 27951 traces, comprising 200 measurements with two user-defined horizons, the tracing of horizons was completed in 12 seconds using double the amount of RAM compared to the size of the processed data. The developed approach does not solve the problem of fault detection but can take their presence into account due to the ability to set fixed borders tracked by the user as function argument.

### Keywords

Seismic exploration, reflection horizons, stratigraphic boundaries, mutual correlation function, process automation, Lee algorithm

### For citation

Vlasov A. A., Romanov N. S. Development of a software module for fast tracing of horizons in a seismic cube using Lee algorithm *Vestnik NSU. Series: Information Technologies*, 2025, vol. 23, no. 1, pp. 19–32 (in Russ.) DOI 10.25205/1818-7900-2025-23-1-19-32

## Введение

Анализ сейсмических данных позволяет обнаружить перспективные зоны для бурения в процессе разведки залежей углеводородов. Такой подход основан на отражении сейсмических волн от границ слоев с разными акустическими свойствами (отражающих горизонтов). Изучая время прохождения через слои и амплитуду отраженных волн, можно получить информацию о составе и структуре земной коры до 20 км в глубину. Данный метод получил название – сеймостратиграфия [1]. Можно выделить три этапа проведения сеймостратиграфического исследования [2]:

- 1) полевые работы – установка датчиков и источников акустических волн, регистрация сейсмических сигналов;
- 2) первичная обработка данных – фильтрация шумов, выравнивание амплитуд, коррекция временных задержек и расчет набора трансформированных сигналов;
- 3) интерпретация – стратиграфическая привязка, корреляция отражающих горизонтов, построение глубинно-скоростных моделей (ГСМ) и т. д.

В рамках данной работы рассматривается автоматизация этапа интерпретации, а именно определение отражающих горизонтов в трехмерном сейсмическом кубе.

В текущих реалиях вручную интерпретация не ведется. Существует ряд автоматических и полуавтоматических проприетарных инструментов, автоматизирующих некоторые шаги интерпретации, например HorizonCube – инструмент от компании dGB Earth Sciences [3]. Тем

не менее доля необходимого ручного труда и редактирования, необходимого для построения модели, согласно стратиграфическим принципам, остается обязанностью эксперта и занимает существенную часть времени. Интерпретация трехмерных сейсмических данных может занимать много времени и давать недетерминированные результаты, так как обработка данных выполняется в условиях большой неопределенности, ввиду чего два эксперта, работая независимо, неизменно получают существенно различающиеся результаты. Машинное обучение не так давно вошло в данную область и уже способно автоматизировать часть рутинных задач эксперта (например, выделение разломов, построение ГСМ), хотя и имеет ряд своих проблем: необходимость большого количества данных для обучения моделей; проблема объяснимости полученного решения; уникальность каждого отдельно взятого геологического объекта, что неизменно может приводить к получению несогласованных моделей [4; 5].

На стадии интерпретации важно добиться согласованности полученной модели, чтобы снизить риски, связанные с инвестициями в будущий проект, путем более достоверной оценки условий освоения месторождения и объема извлекаемых полезных ископаемых [6]. Согласованность достигается за счет учета как можно большего числа стратиграфических принципов. Стратиграфия – это наука, которая фокусируется на исследовании структуры осадочных образований, их возраста и условий формирования [7]. Любые исследования в области стратиграфии подчиняются 5 основным принципам (правилам осадконакопления) [8; 9]:

- принцип первичной горизонтальности (англ. original horizontality) – слои осадочных горных пород изначально осаждаются в виде горизонтальных слоев;
- принцип суперпозиции (англ. superposition) – в ненарушенной стратиграфической последовательности осадочные слои располагаются в порядке от самых старых внизу к самым молодым сверху;
- принцип включений (англ. inclusions) – если один геологический объект (например, минерал или фрагмент породы) содержится внутри другого объекта, то включение старше, чем сама порода, в которой оно находится;
- принцип пересечения (англ. cross-cutting relationship) – геологический объект, который пересекает или нарушает другой объект (например, разлом или инъекционная порода), является более молодым, чем тот объект, который он пересекает;
- принцип преемственности фаунистических сообществ (англ. faunal succession) – существует последовательность, в которой организмы развивались по отношению к геологическому времени, а значит, наличие определенных ископаемых может помочь установить возраст этих слоев.

Задача интерпретации относится к типу обратных задач, ввиду чего существует некоторое множество возможных решений [10; 11]. В ходе анализа специалист оценивает сигналы рядом стоящих трасс, имеющих похожие сигналы. Одна и та же порода будет иметь одинаковые по амплитуде сигналы, но также одинаковые сигналы могут иметь и разные модели (объекты исследования). Эксперт в области интерпретации сейсмических данных учитывает все стратиграфические принципы, свое субъективное мнение и накопленный опыт при построении модели, чтобы получить как можно более правдоподобное решение и более точно определить модель, по которой построен геологический объект. Разработанные алгоритмы, представленные в данной работе, автоматизируют процесс построения согласованной сеймостратиграфической модели, учитывая некоторые из этих правил. Существует ряд сложностей устройства геологических объектов, частично вытекающих из описанных принципов – наличие нарушений сплошности горных пород, так называемых разломов и прочие факторы неоднородности [5]. Описанный подход соблюдает принцип первичной горизонтальности и принцип суперпозиции, но не берет во внимание остальные правила и не решает задачу детектирования разломов, хотя эта проблема является открытой и важной для решения практических задач истолкования сейсмических сигналов. Принцип преемственности фаунистических сообществ не поддается автоматизации ввиду отсутствия необходимой информации. Несмотря на такие ограничения

со стороны алгоритмов, пользователь способен передать обнаруженные им разломы (опорные границы) в виде аргументов функции обработки. Такая априорная информация будет учтена и модель будет построена в соответствии с заданными пользователем ограничениями, что приведет к исключению множества недостоверных решений с точки зрения эксперта (интерпретатора).

В основе разработанного подхода лежат использование корреляционного планшета (аналог планшета Жековского, только вместо каротажных данных – сейсмические) и алгоритмы стратиграфической корреляции разрезов скважин (выделение наиболее похожих областей двух сигналов), основанные на вычислении квадрата расстояния между двумя функциями, в качестве функции метрики [12]. Данный подход хорошо показал себя на двумерных данных, но чаще всего при построении сеймостратиграфических моделей необходимо строить трехмерные модели, ввиду чего появляется больше данных и трудностей интерпретации.

В рамках данной работы предполагается разработка алгоритма, который позволяет на основании априорной информации производить автокорреляцию трехмерных сейсмических трасс и выделять отражающие горизонты подповерхностного слоя. Для определения степени похожести пары близлежащих трасс рассматриваются их относительные сдвиги по времени, благодаря чему можно получить матрицу похожести точек одной трассы на другую и, при помощи волнового алгоритма трассировки, найти значения корреляции между этой парой трасс. Полученный результат должен быть детерминирован и легко объясним. Как было указано ранее, предложенный подход не подразумевает решения проблемы детектирования разломов.

Цель работы – автоматизировать процесс прослеживания (взаимозаменяемый термин – трассировки) отражающих горизонтов в трехмерных сейсмических данных на основе априорной информации и ограничений, выставленных пользователем, соблюдая два стратиграфических принципа (первичной горизонтальности и суперпозиции), применяя корреляционный планшет, волновой алгоритм и алгоритм динамической трансформации временной шкалы (DTW) в ходе попарной корреляции сейсмических трасс.

Для достижения данной цели были пройдены следующие этапы:

1. Разработка функции взаимной корреляции, учитывающей априорно заданные отражающие горизонты и указанные принципы стратиграфии, используя алгоритм DTW.
2. Разработка алгоритма волновой трассировки для поиска пути в матрице корреляции.
3. Разработка функции построения графа корреляции в сейсмическом кубе, где вершины – трассы, а ребра – отображения из одной трассы в другую.
4. Реализация построения остовного дерева с оставлением ребер между максимально похожими друг на друга трассами.
5. Внедрение разработанных алгоритмов в программный комплекс W-SEIS для визуализации результатов и отладки.

## 1. Метод построения корреляционной модели

Автокорреляция пары трасс (скважин) осуществляется путем сравнения двух сейсмических кривых: для любой пары значений сейсмических трасс,  $xi \in x$  и  $uj \in y$ , вычисляется специальная двумерная функция, характеризующая степень различия пары сигналов. Их похожесть рассматривается в особых «окнах», установленных в каждом из этих сигналов, где размер окна задается параметром –  $r\_idx$  (размер окна равен  $2*r\_idx + 1$ ). В качестве функции оценки похожести пары сейсмических сигналов была выбрана функция свертки:  $fij=i,j = -r\_idxr\_idxijj$ , где  $xi, i \in [0, h1)$ ,  $uj \in [0, h2)$ ,  $h1$  и  $h2$  – глубины залегания сигнала первой трассы и второй. Преимущество указанной функции в том, что она может быть быстро вычислена на процессоре с помощью векторных инструкций. В конце производится нормализация значения с сохранением знака:  $res = sgn(f)*fn$ . Чем больше значение, тем более похожи участки сигналов.

Расчет функции свертки (*res*) имеет смысл только для пар похожих по форме отрезков сейсмической кривой, и близлежащие трассы, как правило, имеют очень похожие формы сигналов с некоторым временным смещением, а значит, применять формулу *res* на паре соседних трасс можно, если рассчитывать *res* в ограниченном диапазоне индексов (окне). Размер этого окна задается параметром *max\_shift*. На расстоянии, не превышающим *max\_shift* между индексами *i* и *j*, производится расчет функции *res*.

Таким образом, производится перебор всех значений *xi* и *yj* в заданном интервале, получаем для них значения различия. Имеем матрицу, где каждому сигналу первой трассы противопоставлен сигнал второй трассы и для каждой такой пары рассчитана мера различия. Такая матрица похожа на планшет Жековского, только вместо каротажных данных – сейсмические [13]. Также в терминологии DTW, в англоязычных статьях распространено название *cost matrix* [14; 15]. Далее эта матрица будет называться *корреляционный планшет*. Такое представление позволяет рассмотреть прямые, соединяющие сигналы двух трасс на разных глубинах и оценить похожесть сигналов. На рис. 1 представлен пример вычисленного корреляционного планшета для пары сейсмических трасс разного размера с прослеженной корреляционной линией.

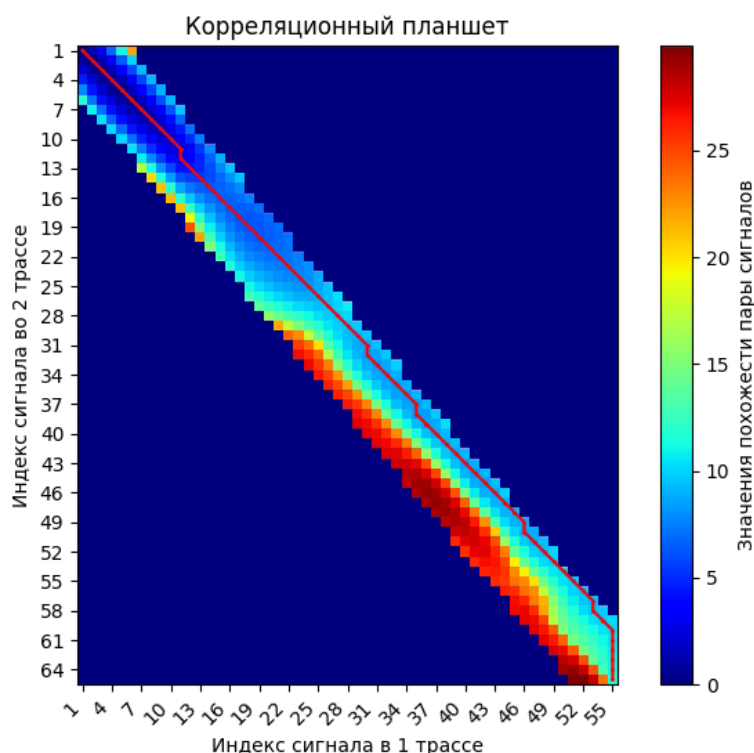


Рис. 1. Корреляционный планшет с прослеженной корреляционной линией.

Чем меньше значения, тем более похожи два сигнала

Fig. 1. Correlation tablet with traced correlation line. The smaller the values, the more similar the two signals are

Далее необходимо получить корреляционную модель. Корреляционная модель – это набор точек матрицы (пары координат *x* и *y*), образующих кривую, соединяющую два угла корреляционного планшета. В нашем случае не допускается возможность пересечения границ подповерхностных слоев – соблюдаются два правила осадконакопления (принцип первичной горизонтальности и принцип суперпозиции). Точки кривой выбираются таким образом, чтобы различия между сравниваемыми скважинами были минимальны. Задача сводится к нахожде-

нию кратчайшего пути, для чего используется волновой алгоритм Ли [16]. Результатом работы алгоритма является кривая, точки которой представляют оптимальное отображение точек одной глубины первой сейсмической кривой к точкам второй. По таким отображениям производится совмещение пары скважин и трассировка стратиграфических границ.

Алгоритм Ли чувствителен к отрицательным значениям во взвешенном графе, поэтому прежде чем его применить, необходимо произвести нормализацию значений в корреляционном планшете. Пусть  $max\_val$  – максимальное значение похожести среди всех положительных значений в матрице; минимальное положительное число, если положительных значений нет. Нормализуем все значения матрицы:  $cur\_val[i][j] = max\_val - cur\_val[i][j] + 1\epsilon-100$ . Таким образом, получаем, что минимальные значения матрицы отражают наибольшую похожесть пары сигналов; отрицательные сигналы отсутствуют, что дает возможность использовать волновой алгоритм; поиск минимального пути покажет отображения одной трассы на другую.

На рис. 2 изображено соответствие сигналов пары сейсмических кривых в более простом для восприятия человеком пространстве.

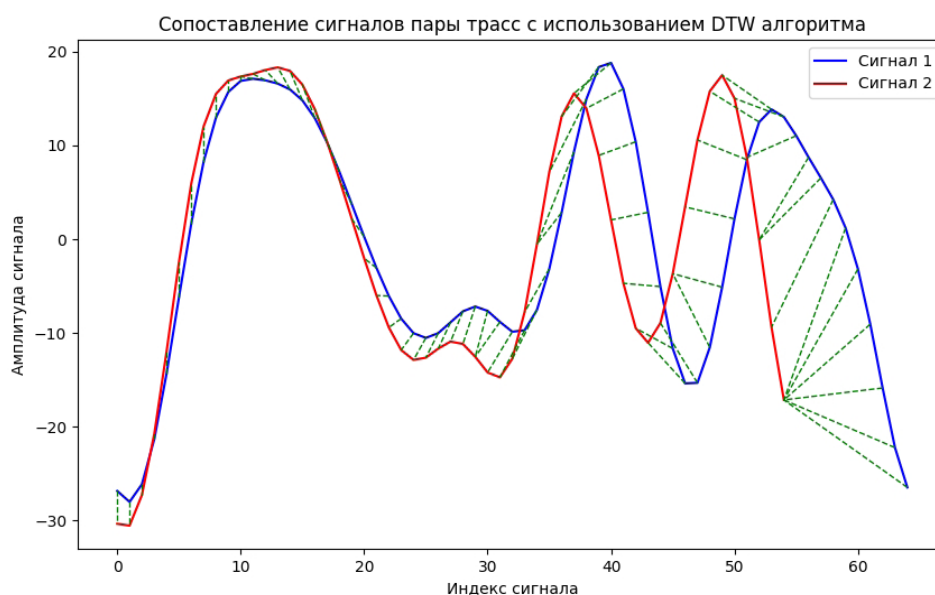


Рис. 2. Сопоставление сигналов пары трасс разной длины с использованием алгоритма трансформации временной шкалы  
Fig. 2. Comparison of signals of pair seismic traces with different size using the DTW algorithm

Теперь можно получить величину различия двух сейсмических кривых следующим образом:  $diff = min\_path_{length} - max\_val$ , где  $min\_path$  – длина наименьшего пути, а  $path\_length$  – число точек в кратчайшем пути.

Корреляционная модель, полученная вышеописанным методом, неустойчива. Всегда будет существовать некоторый набор кривых, разных по форме, но очень близких по значениям меры различия [11]. Если внести даже несущественные корректировки во входные параметры алгоритма, результат прослеживания может получиться совершенно другой, и эксперт может быть с ним не согласен. Поэтому в разработанные алгоритмы была добавлена возможность ручной фиксации отражающих горизонтов путем передачи соответствующего параметра. Этот параметр заставляет волновой алгоритм Ли проходить через некоторые точки (ручное задание весов), даже если они не считаются оптимальными.

Временная сложность описанного алгоритма равна  $O((M2*(2*r\_idx-1)+M2)*((n-1)*m+(m-1)*n))$ , где  $M$  – максимальный размер сейсмической кривой (число сигналов),  $n$  –

число трасс вдоль оси  $x$ ,  $m$  – число трасс вдоль оси  $y$ . Первый множитель описывает процесс автокорреляции пары трасс и состоит из времени, затрачиваемого на расчет операции свертки для каждой пары сигналов двух трасс (в худшем случае – когда  $max\_shift \geq M$ ) и второго слагаемого – поиск пути при помощи волнового алгоритма. Второй множитель появился ввиду того, что операция автокорреляции выполняется столько раз, сколько пар сейсмических кривых будет подвергнуто корреляции.

Пространственная сложность:  $O(N*M + ((n-1)*m + (m-1)*n)*2M)$ , где  $N*M$  – сигналы для всех трасс,  $(n-1)*m + (m-1)*n$  – число ребер (пар сейсмических трасс),  $2M$  – отображение каждого сигнала первой трассы на каждый сигнал второй и обратно.

## 2. Метод прослеживания отражающих горизонтов

Описанный ранее метод применяется попарно на всех трассах, находящихся не далее одного индекса. В итоге получаем граф, где каждая вершина представляет собой одну сейсмическую кривую и связана ребрами с соседями. Ребро хранит информацию о схожести ( $diff$ ) и все отображения точек одного сигнала на второй и обратно.

Для прослеживания стратиграфических границ необходимо обойти все вершины полученного графа и назначить всем глубинам свой индекс границы так, чтобы получилось много непрерывных плоскостей. В текущей реализации на расположение индексов границ в одной трассе влияет от 2 до 4 ее соседей, соответственно, столько же и ребер. Выбор того ребра, по которому будет определяться конечное множество плоскостей, должно исходить из соображений выбора пары наиболее похожих друг на друга сейсмических сигналов. И чтобы осуществить такой обход, исходный граф приводится к виду минимального остовного дерева, т. е. выбирается минимальное число ребер с минимальным возможным весом так, чтобы можно было обойти все вершины графа. Для этого применяется алгоритм Прима [17]. Опорная точка, с которой будет осуществляться обход дерева, задается как параметр алгоритма.

На рис. 3 представлен результат работы алгоритма построения остовного дерева на сейсмическом кубе с размерностью 200 трасс.

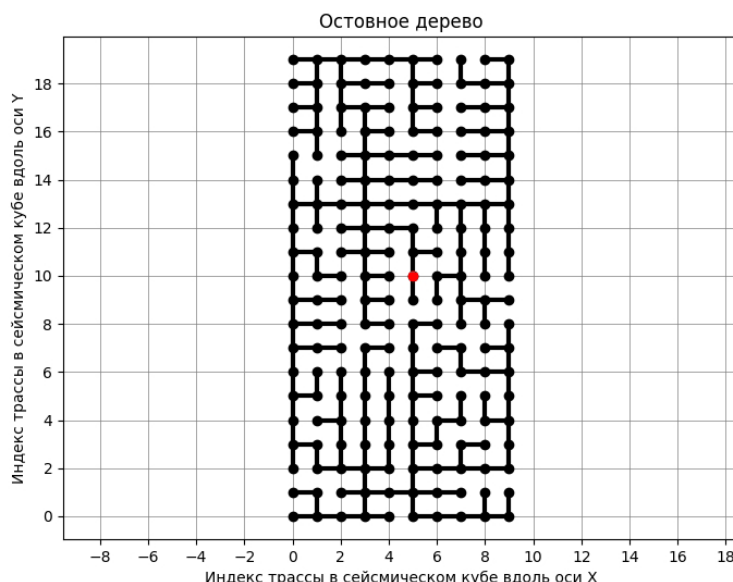


Рис. 3. Остовное дерево в сейсмическом кубе на 200 трасс.

Красным выделена опорная трасса. Глубина дерева – 38

Fig. 3. Spanning tree in a seismic cube with 200 traces.

The reference trace is highlighted in red. The depth of the tree is 38

На сейсмическом кубе, размером 27951, при выбранной в центре опорной трассе, глубина дерева достигает 1102.

Далее осуществляется обход в глубину в полученном минимальном остовном дереве, начиная с заданной в качестве аргумента опорной трассы. Определять отражающие горизонты можно разными способами, но общий принцип один: сигналам опорной трассы назначаются некоторые номера границ, например, минимальные номера обозначают самые молодые границы, а наибольшие – самые древние по времени геологического формирования. Также можно задавать через параметры алгоритма толщину этих границ (количество сигналов, которыми будет обозначена одна граница) и промежутки между парой соседних границ. Далее, на каждом шаге алгоритма, происходит распространение этих номеров границ на соседние трассы (для каждого сигнала каждой трассы известно, куда его необходимо отображать на соседней трассе), основываясь на результатах полученной ранее корреляционной модели.

В результате будет получено множество поверхностей, каждая из которых представляет собой выделенную стратиграфическую границу.

Асимптотическая сложность прослеживания отражающих горизонтов, когда корреляционная модель уже построена:  $O(E + V \log V + n * m * M)$ , где  $E$  и  $V$  – количество ребер и вершин в исходной корреляционной модели соответственно;  $E + V \log V$  – построение остовного дерева;  $n * m * M$  – трассировка отражающих горизонтов, заполнение поверхностей.

### 3. Внедрение разработанного подхода в программный комплекс интерпретации сейсмических и скважинных данных W-SEIS

Разработанные алгоритмы были интегрированы в программный комплекс интерпретации сейсмических и скважинных данных W-SEIS, а конкретно, в модуль Desmana (модуль, предназначенный для работы с сейсмическими данными) [18]. В качестве тестового сейсмического куба использовался куб размером 27951 (231 \* 121) трасс и двумя отражающими горизонтами, прослеженными экспертом.

На рис. 4 изображен пользовательский интерфейс для задания параметров, таких как выбор подхода корреляции, размер диагонали в корреляционном планшете, выбор сейсмического куба, выбор зафиксированных границ, задание глубинных индексов и пр. и запуска разработанного алгоритма. Также ряд параметров для настройки работы алгоритмов был вынесен в отдельный конфигурационный файл, изображенный на рис. 5. Он позволяет задать размер выбранного подмножества трасс из всего куба, опорную трассу, размер границы и отступ между границами для визуализации и радиус фиксации точки.

Осуществлено прослеживание стратиграфических границ на рабочей станции с процессором AMD Ryzen 7 4800H и оперативной памятью 16 Гб с заданными в виде аргументов ограничениями – пара прослеженных пользователем границ, между которыми проводилось исследование; размер диагонали в корреляционном планшете, равный 5 индексам; расстояние между заданными границами в 200 сигналов; опорная точка выбрана в центре сейсмического куба. Такой эксперимент на указанной рабочей станции занимает 12 секунд. На рис. 6 показан результат прослеживания границ при помощи описанного алгоритма в Desmana.

Нетрудно заметить, что изначально поверхность отражающего горизонта прослеживается корректно (если наблюдать изменение из левого края, где была расположена опорная точка, а значит, и точка начала обхода дерева), и это происходит до тех пор, пока производится попарная корреляция тех трасс, которые изображены на рис. 6 (трасс вдоль видимой прямой), но так как в остовном дереве ребра не обязательно будут находиться вдоль одной прямой, решение может пройти существенную часть куба, прежде чем оказаться в соседней трассе этой прямой. В результате этого процесса возникает ошибка, которая сначала выражается в виде смещенных

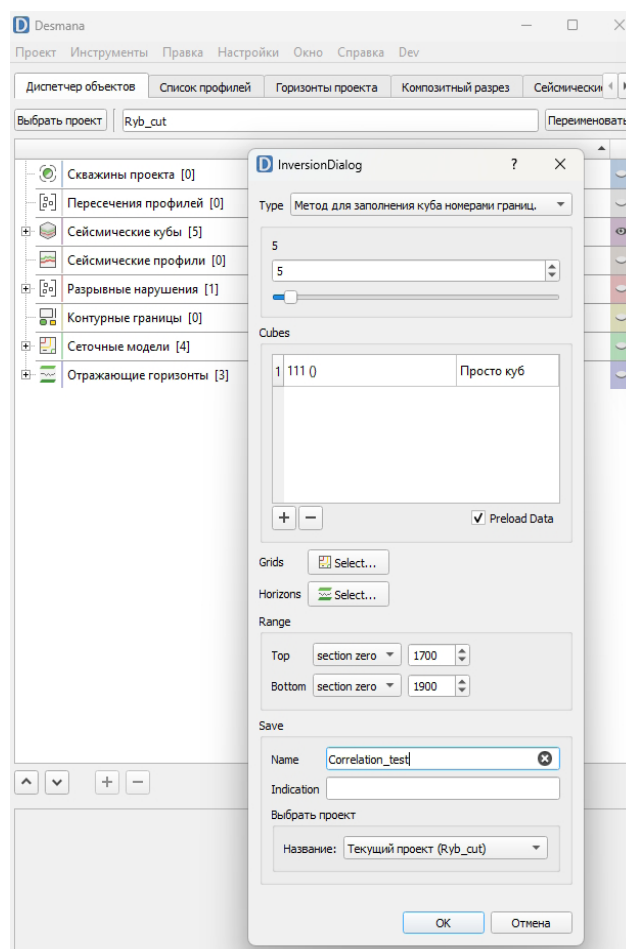


Рис. 4. Тестовый интерфейс для отладки плагина в Desmana  
Fig. 4. Test plugin interface for debugging implementation in Desmana

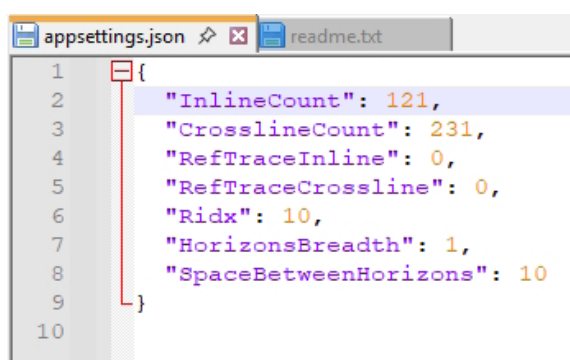


Рис. 5. Параметры, задаваемые через конфигурационный файл: InlineCount – число трасс вдоль оси  $x$ ; CrosslineCount – число трасс вдоль оси  $y$ ; RefTraceInline/Crossline – координаты опорной трассы; Ridx – радиус фиксации точки; HorizonsBreadth – размер границы; SpaceBetweenHorizons – количество индексов между парой ближайших границ на опорной трассе

Fig. 5. Arguments set through the configuration file: InlineCount is the number of traces along the  $x$ -axis; CrosslineCount is the number of traces along the  $y$ -axis; RefTraceInline/Crossline is coordinates of the reference trace; Ridx is a radius of point fixation; HorizonsBreadth is a size of the borders; SpaceBetweenHorizons is a number of indexes between the pair of nearest borders on the reference trace

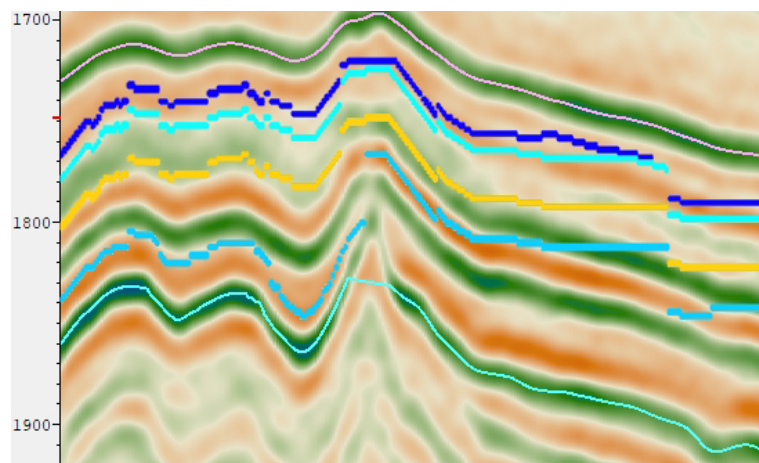


Рис. 6. Визуализация прослеженных поверхностей методом построения остоного дерева, отражающих стратиграфические границы. Самая верхняя и нижняя границы заданы пользователем  
Fig. 6. Visualization of traced surfaces using the minimum spanning tree method, reflecting stratigraphic boundaries.  
The uppermost and lowermost boundaries are defined by the user

индексов отражающих горизонтов, а затем все больше нарастает и вовсе перестает отражать закономерности устройства исследуемого объекта.

Становится ясно, что для достижения устойчивого и согласованного результата необходимо опираться не только на результат корреляции пары соседних трасс, но и на предыдущие результаты корреляции.

Чтобы побороться с этой проблемой, был опробован иной способ прослеживания отражающих горизонтов.

#### **4. Метод прослеживания отражающих горизонтов, основанный на корреляции с опорной трассой**

Для поддержания согласованности модели было решено произвести корреляцию опорной трассы со всеми остальными в соответствии с описанным методом построения корреляционной модели. На основании полученной модели уже можно произвести трассировку отражающих горизонтов, без необходимости построения остоного дерева.

Далее, действуя по общему принципу прослеживания границ, назначаем номера границ различным точкам на опорной сейсмической кривой, а дальше обходим каждую вершину графа по ребру, которое исходит из опорной вершины и дает представление об отображении каждой точки опорной прямой на выбранную. Таким образом протягиваются номера границ из опорной трассы на все остальные и снова получают поверхности отражающих горизонтов.

Алгоритм был опробован на указанной ранее рабочей станции, исполнение заняло 7,5 секунды. Результат работы такого подхода изображен на рис. 7. На рисунке можно заметить, что принцип первичной горизонтальности и принцип суперпозиции не нарушаются и соблюдаются закономерности устройства исследуемого объекта. Такое решение обратной задачи можно считать корректным, если не брать во внимание остальные факторы и ограничения.

Недостатком данного подхода является игнорирование зависимостей каждой выбранной сейсмической трассы с ее соседями, по сути игнорирование априорного знания о гладкости границ, что иногда приводит к резким «скачкам» кривой, отражающей границу.

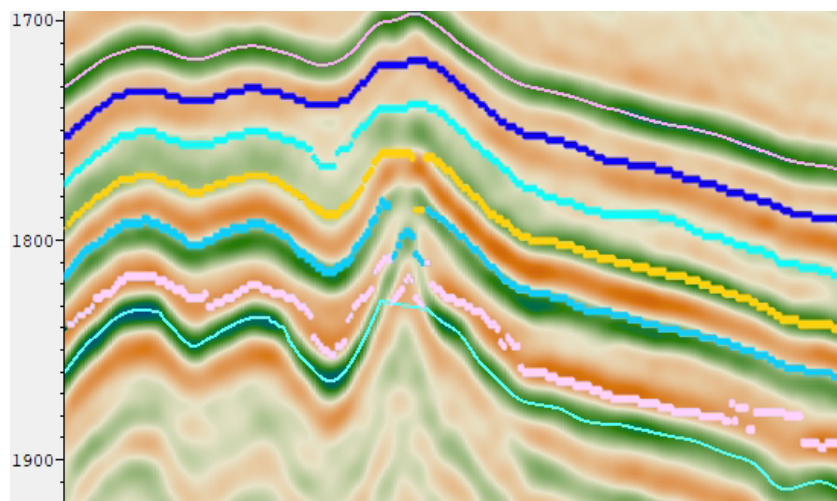


Рис. 7. Визуализация прослеженных поверхностей, отражающих стратиграфические границы, методом корреляции всех трасс с опорной. Самая верхняя и нижняя границы заданы пользователем  
Fig. 7. Visualization of traced surfaces reflecting stratigraphic boundaries by correlating all trails with a reference one.  
The uppermost and lowermost boundaries are defined by the user

Асимптотическая сложность такого подхода значительно меньше в сравнении с попарной корреляцией соседей, так как здесь мы сравниваем только опорную трассу со всеми остальными:  $O((M2*(2*r\_idx-1)+M2)*(m*n))$ .

## Выводы

В настоящее время изложенный подход реализован в двух вариантах и интегрирован в программный комплекс W-SEIS, модуль Desmana в виде dll-плагина. Оба алгоритма хорошо распараллеливаются и могут быть эффективно исполнены на многоядерных архитектурах вычислительных устройств, например возможна реализация отдельного специализированного вычислительного модуля на основе FPGA, что позволит еще на один или два порядка ускорить вычисления и решать задачу для пользователя в интерактивном режиме. Второй алгоритм выполняется на 60 % быстрее в сравнении с первым и может быть использован экспертом для получения модели, учитывающей принцип первичной горизонтальности и суперпозиции, сохраняя закономерности устройства исследуемого объекта, но важно понимать, что на выбор решения не влияют соседние сейсмические кривые, а значит, эксперту необходимо будет проверить и существенно дополнить полученную модель, приняв во внимание взаимное влияние соседних трасс для повышения согласованности модели. Первый алгоритм в текущем виде требует доработки и не может быть использован для решения реальных задач ввиду сильно разрастающейся ошибки.

Предложенные подходы опираются на два стратиграфических принципа при составлении модели, ввиду чего данное решение является, скорее, вспомогательным инструментом для эксперта. Полезность второго алгоритма в том, что существует возможность задания фиксированных точек (границ), которые будут учтены при построении модели, что позволяет ускорить работу эксперта, предлагая ему множество возможных решений, которые ему необходимо проанализировать.

Дальнейшее развитие идей, описанных в этой работе, следует направить на разработку третьего алгоритма, который включит в себя две идеи, описанные здесь: учет влияния соседних трасс на общее решение и сверка результатов с предыдущими шагами, выбор опорных то-

чек для борьбы с разрастающейся ошибкой. Также для увеличения согласованности получаемых моделей можно рассматривать влияние трасс-соседей не только вдоль двух координатных осей, но и в некоторых пространственных трансформациях, что могло бы поспособствовать учету принципа включений и принципа пересечений.

### Список литературы

1. **Симм Р., Бэкон М.** Амплитуда сейсмических колебаний: руководство для переводчика. Кембридж; Нью-Йорк: Изд-во Кембридж. ун-та, 2014.
2. **Мондол Н. Х.** Сейсморазведка // *Petroleum Geoscience*. 2010. С. 375–402. Doi: [https://doi.org/10.1007/978-3-642-02332-3\\_17](https://doi.org/10.1007/978-3-642-02332-3_17).
3. **Де Гроот П.** Глобальные методы интерпретации сейсмических данных достигают совершенства // *Расширенные тезисы ASEG*, декабрь 2013. Т. 2013. № 1. С. 1–4. Doi: <https://doi.org/10.1071/aseg2013ab331>.
4. **Гэн З., Ву Х., Ши Ю., Фомел С.** Оценка относительного геологического времени с использованием глубокой сверточной нейронной сети, август 2019 г., Doi: <https://doi.org/10.1190/segam2019-3214459.1>.
5. **Вэй Х.-Л. и др.** Обнаружение сейсмических разломов с использованием сверточных нейронных сетей с потерей фокуса // *Компьютеры и науки о земле*. 2022. Т. 158. С. 104968, Doi: <https://doi.org/10.1016/j.cageo.2021.104968>.
6. **Де Гроот П., Хак А., де Брюин Г., Хемстра Н., Бедфорд Дж.** Куб горизонта: пошаговое изменение в интерпретации сейсмических данных // *The Leading Edge*. 2010. Т. 29, № 9. С. 1048–1055, Doi: <https://doi.org/10.1190/1.3485765>.
7. **Эдуардо.** Стратиграфия: эволюция концепции. Springer eBooks, 2006, с. 3–19, doi: [https://doi.org/10.1007/1-4020-2763-x\\_1](https://doi.org/10.1007/1-4020-2763-x_1).
8. **Кравиц Дж.** Геоисторическая стрела времени: от стратиграфических принципов Стено к гипотезе Больцмана о прошлом // *Journal of Geoscience Education*. 2014. Vol. 62, № 4. Р. 691–700, doi: <https://doi.org/10.5408/13-107.1>.
9. **Маклеод Н.** Принципы стратиграфии. Нанкинский университет, 2005.
10. **Лапковский В., Истомин А., Конторович В., Бердов В.** Корреляция каротажных данных скважин как многомерная задача оптимизации // *Философия и физика*. 2015. Т. 56, № 3, Doi: <https://doi.org/10.15372/gig20150309>.
11. **Ланда Э., Трейтель С.** Сейсмическая инверсия: что это такое и чем она не является // *The Leading Edge*. 2016. Т. 35, № 3. С. 277–279, doi: <https://doi.org/10.1190/tle35030277.1>.
12. **Лапковский В. В.** Непрерывная сейсмостратиграфическая модель как основа структурной интерпретации разрезов МОГТ // *Технологии сейсморазведки*. 2012. № 4. С. 33–39.
13. **Лапковский В. В., Шмелев Н. Е., Лунев Б. В.** Составное изображение скважины по данным каротажа: технология создания и применения // *Нефтегазовая геология, Теория и практика*. 2018. Т. 13, № 2, doi: [https://doi.org/10.17353/2070-5379/11\\_2018](https://doi.org/10.17353/2070-5379/11_2018).
14. **Динамическое искривление времени. Поиск информации по музыке и движению**, 2007. С. 69–84, 2007, doi: [https://doi.org/10.1007/978-3-540-74048-3\\_4](https://doi.org/10.1007/978-3-540-74048-3_4).
15. **Херрманн М., Чанг Вэй Тан, Уэбб Г. И.** Параметризация функции затрат при динамическом искажении времени применительно к классификации временных рядов // *Интеллектуальный анализ данных и обнаружение знаний*. 2023. Т. 37, № 5. С. 2024–2045, doi: <https://doi.org/10.1007/s10618-023-00926-8>.
16. **Ли С. Ю.** Алгоритм для соединения путей и его приложения // *IRE Transactions on Electronic Computers*. 1961, EC-10 (3). С. 346–365.
17. **Петти С., Рамачандран В.** Алгоритм оптимального минимального остовного дерева // *Журнал АСМ*. 2002. Т. 49, № 1. С. 16–34.

18. **Канаков М. С., Лапковский В. В., Конторович В. А., Канакова К. И., Шорохов Ю. С.** Программное обеспечение для интерпретации данных сейсморазведки и бурения скважин W-SEIS // Российский журнал геофизических технологий. 2022. № 2. С. 77–91, doi: <https://doi.org/10.18303/2619-1563-2022-2-77>.

## References

1. **Simm R., Bacon M.** Seismic amplitude: an interpreter's handbook. Cambridge; New York: Cambridge University Press, 2014.
2. **Mondol N. H.** Seismic Exploration, Petroleum Geoscience, 2010, pp. 375–402, doi: [https://doi.org/10.1007/978-3-642-02332-3\\_17](https://doi.org/10.1007/978-3-642-02332-3_17).
3. **De Groot P.,** Global Seismic Interpretation Techniques Are Coming of Age. ASEG Extended Abstracts, Dec. 2013, vol. 2013, no. 1, pp. 1–4, doi: <https://doi.org/10.1071/aseg2013ab331>.
4. **Geng Z., Wu X., Shi Y., Fomel S.,** Relative geologic time estimation using a deep convolutional neural network, Aug. 2019, doi: <https://doi.org/10.1190/segam2019-3214459.1>.
5. **Wei X.-L.** et al. Seismic fault detection using convolutional neural networks with focal loss. *Computers & Geosciences*, Jan. 2022, vol. 158, p. 104968, doi: <https://doi.org/10.1016/j.cageo.2021.104968>.
6. **De Groot P., Huck A., de Bruin G., Hemstra N., Bedford J.** The horizon cube: A step change in seismic interpretation! *The Leading Edge*, Sep. 2010, vol. 29, no. 9, pp. 1048–1055, doi: <https://doi.org/10.1190/1.3485765>.
7. **Eduardo.** Stratigraphy: Evolution of a Concept, Springer eBooks, Aug. 2006, pp. 3–19, doi: [https://doi.org/10.1007/1-4020-2763-x\\_1](https://doi.org/10.1007/1-4020-2763-x_1).
8. **Kravitz G.** The Geohistorical Time Arrow: From Steno's Stratigraphic Principles to Boltzmann's Past Hypothesis. *Journal of Geoscience Education*, 2014, vol. 62, no. 4, pp. 691–700, doi: <https://doi.org/10.5408/13-107.1>.
9. **MacLeod N.** Principles of stratigraphy. Nanjing University, 2005.
10. **Lapkovsky V., Istomin A., Kontorovich V., Berdov V.** CORRELATION OF WELL LOGS AS A MULTIDIMENSIONAL OPTIMIZATION PROBLEM. *Геология и геофизика*, 2015, vol. 56, no. 3, doi: <https://doi.org/10.15372/gig20150309>.
11. **Landa E., Treitel S.** Seismic inversion: What it is, and what it is not. *The Leading Edge*, Mar. 2016, vol. 35, no. 3, pp. 277–279, doi: <https://doi.org/10.1190/tle35030277.1>.
12. **Lapkovsky V. V.** Continuous seismostratigraphic model as the basis for the structural interpretation of MOGT sections // *Seismic Exploration Technologies*, No. 4, pp. 33–39, Jan. 2012. (in Russ.)
13. **Lapkovsky V. V., Shmelev N. E., Lunev B. V.** Composite well image from log data: technology of creation and application. *Neftegazovaya Geologiya Teoriya I Praktika*, Jan. 2018, vol. 13, no. 2, doi: [https://doi.org/10.17353/2070-5379/11\\_2018](https://doi.org/10.17353/2070-5379/11_2018).
14. **Dynamic Time Warping.** Information Retrieval for Music and Motion, pp. 69–84, 2007, doi: [https://doi.org/10.1007/978-3-540-74048-3\\_4](https://doi.org/10.1007/978-3-540-74048-3_4).
15. **Herrmann M., Chang Wei Tan, Webb G. I.** Parameterizing the cost function of dynamic time warping with application to time series classification. *Data Mining and Knowledge Discovery*, Apr. 2023, vol. 37, no. 5, pp. 2024–2045, doi: <https://doi.org/10.1007/s10618-023-00926-8>.
16. **Lee C. Y.** An Algorithm for Path Connections and Its Applications, *IRE Transactions on Electronic Computers*, 1961, EC-10 (3), pp. 346–365.
17. **Pettie S., Ramachandran V.** An optimal minimum spanning tree algorithm. *Journal of the ACM*, Jan. 2002, vol. 49, no. 1, pp. 16–34.

18. **Kanakov M. S., Lapkovsky V. V., Kontorovich V. A., Kanakova K. I., Shorohov Y. S.** W-SEIS interpretation software for seismic and well drilling data. *Russian Journal of geophysical technologies*, Dec. 2022, no. 2, pp. 77–91, doi: <https://doi.org/10.18303/2619-1563-2022-2-77>.

### Информация об авторах

**Власов Александр Александрович**, кандидат технических наук, доцент Новосибирского государственного университета, инженер Института автоматизации и электрометрии СО РАН

**Романов Никита Сергеевич**, магистрант

### Information about the Authors

**Alexander A. Vlasov**, Ph.D., Associate Professor – Novosibirsk State University, Engineer – Institute of Automation and Electrometry SB RAS

**Nikita S. Romanov**, Master's Student

*Статья поступила в редакцию 19.02.2025;  
одобрена после рецензирования 14.03.2025; принята к публикации 14.03.2025*

*The article was submitted 19.02.2025;  
approved after reviewing 14.03.2025; accepted for publication 14.03.2025*

Научная статья

УДК 004.421.5

DOI 10.25205/1818-7900-2025-23-1-33-45

## **Сравнительный анализ статистических тестов на основе теста сжатия данных и стандартных тестов для оценки случайности генераторов случайных чисел**

**Йешевас Гетачеу Лулу**

Новосибирский государственный университет  
Новосибирск, Россия

j.lulu@g.nsu.ru, <https://orcid.org/0009-0006-8054-9846>

### *Аннотация*

В работе представлен подробный сравнительный анализ статистических тестов, использующих как современные компрессоры данных, так и стандартные статистические методы для оценки случайности генераторов случайных чисел (ГСЧ). Цель исследования – всесторонне оценить эффективность и результативность этих тестов при определении качества выходных последовательностей ГСЧ.

Методы сжатия данных давно признаны эффективными статистическими тестами, причем некоторые из них являются асимптотически оптимальными. В статье сравнивается эффективность этих тестов, основанных на сжатии данных, с традиционными статистическими тестами при оценке свойств случайности ГПЗУ.

Результаты исследования показывают, что эффективность тестов на основе компрессора данных и стандартных статистических тестов практически одинакова. Проведенные эксперименты и анализ, показали, что оба подхода дают сопоставимые результаты при оценке случайности выходных последовательностей ГПЗУ.

### *Ключевые слова*

статистические тесты сжатия данных, генератор случайных чисел, статистический тест Национального института стандартов и технологий, тест TESU01

### *Благодарности*

Автор выражает искреннюю благодарность научному руководителю, профессору Б. Я. Рябко за неоценимую помощь в критическом обзоре эксперимента и содержательные комментарии к рукописи.

### *Для цитирования*

Лулу Й. Г. Сравнительный анализ статистических тестов на основе теста сжатия данных и стандартных тестов для оценки случайности генераторов случайных чисел // Вестник НГУ. Серия: Информационные технологии. 2025. Т. 23, № 1. С. 33–45. DOI 10.25205/1818-7900-2025-23-1-33-45

© Лулу Й. Г., 2025

# Comparative Analysis of Statistical Test based on Data compression methods and standard tests for assessing Randomness of Random Number Generators

Yeshewas Getachew Lulu

Faculty of Information Technology Novosibirsk State University  
Novosibirsk, Russian Federation

j.lulu@g.nsu.ru

## Abstract

This paper presents a detailed comparative analysis of statistical tests utilizing both modern data compressors and standard statistical methods for assessing the randomness of Random number generators (RNG). Our study aims to thoroughly evaluate the efficiency and performance of these tests in determining the quality of RNG output sequences. Data compression techniques have long been recognized as effective statistical tests, with some being asymptotically optimal. We compare the effectiveness of these data compressor-based tests with traditional statistical tests in assessing the randomness properties of RNG.

Our results demonstrate that the efficiency of data compressor tests and standard statistical tests is closely similar. Through rigorous experimentation and analysis, we show that both approaches yield comparable results in evaluating the randomness of RNG output sequences.

## Keywords

Data compression statistical tests, random number generator, National Institute of Standards and Technology statistical test, TESU01 test

## Acknowledgement

I would like to express my sincere gratitude to my supervisor, Professor Ryabko Boris Yakovlevich, for his invaluable assistance with the critical review of the experiment and insightful comments on the manuscript.

## For citation

Yeshewas Getachew Lulu. Comparative Analysis of Statistical Test based on Data compression methods and standard tests for assessing Randomness of Random Number Generators. *Vestnik NSU. Series: Information Technologies*, 2025, vol. 23, no. 1, pp. 33–45 (in Russ.) DOI 10.25205/1818-7900-2025-23-1-33-45

## 1. Introduction

Data compression techniques have been recognized as effective statistical tests since paper [1] was published. It has been demonstrated in the literature that data compression tests are asymptotically optimal [2]. We compare the effectiveness of these data compressor-based tests with other well-known test suites such as NIST and TestU01 in assessing the randomness properties of RNG.

Random number generators stand as fundamental components across a wide spectrum of computational domains, ranging from simulations and cryptography to statistical sampling [3–5]. These methods play a critical role in generating sequences of numbers that mimic randomness, a crucial requirement in various applications. Selecting the appropriate RNG is vital to uphold the integrity and dependability of the generated sequences, as biased or predictable outputs can undermine the validity of computational results and compromise security measures [5].

In response to the imperative need for robust RNG, extensive research efforts have been directed towards developing and evaluating these algorithms. A key aspect of this evaluation involves subjecting RNGs to rigorous testing procedures to ascertain their randomness properties. Numerous statistical test suites have emerged to facilitate this evaluation process, enabling researchers and practitioners to measure the performance of RNGs under different testing methodologies [6].

Among the prominent test suites employed for assessing the randomness of RNGs, the National Institute of Standards and Technology (NIST) test suite [7], TestU01 [8], and data compressor tests [9], DIEHARD [10], CRYPT-X [11], a test based on diffusion characteristic of a block cipher [12],

topological binary test defined by Alcover et al [13]. hold significant prominence. These suites offer a comprehensive battery of statistical tests designed to probe various aspects of randomness, including uniformity, independence, and unpredictability. By subjecting RNG outputs to these tests, researchers can gain insights into the strengths and weaknesses of different algorithms, thereby informing their selection for specific applications [14].

Through rigorous experimentation and analysis conducted in this study, four weak RNG (RANDU, linear congruential generator, linear-feedback shift register, Middle Square Generator) and three strong generators (Mersenne Twister, permuted congruential generator, Blum Blum Shub Generator) were examined. Various file lengths (1 KB, 10 KB, 100 KB, 500KB, 1 MB) with 100 sequences each were utilized.

Therefore, this study aims a comparative analysis of RNGs using standard batteries of tests and test based on modern data compression.

The rest of the paper is organized as follows. In section 2 contains methodology used for evaluation of RNG, In Sections 3 we investigate the experimental setup and testing environment, in section 4 experimental results will be shown, in section 5 discussion and interpretation of experimental results are shown and conclusion will be drawn in final sections.

## 2. Methodology

In this section, we detail the methodologies employed to evaluate the performance of Random number generators (RNGs) through three distinct approaches: The Data Compression Test, the NIST Test Suite, and the TestU01 (Big-Crush) Statistical Test Suite.

### 2.1. Data Compression Test

To assess the randomness of RNG output, we examine its compressibility using modern compression algorithms such as Gzip [15], LZMA2 [16], Brotli [18], Zstd [19], and LZFSE [20]. In this context, we consider an alphabet size of  $|A| = 256$ , corresponding to  $2^8$ , with  $n \log |A|$  representing the length of the file in bits before compression (where  $n$  denotes the length in bytes) [1] [21]. For a significance level of  $\alpha = 0.01$ , the hypothesis regarding randomness ( $H_0$ ) is rejected if the length of the compressed file is equal to or less than  $n \log |A| - 8$  bits [21]. To simplify, considering  $\alpha \leq 2^{-7} = 1/128$ , if the  $n$ -byte file is compressed (meaning the length of the encoded file is  $n - 1$  bytes or fewer), it is deemed non-random, leading to the rejection of  $H_0$ .

Flow Chart Approach: The Data Compression Test approach typically involves the following steps:

1. **Generation of RNG Sequences:** First, RNG sequences are generated using selected algorithms.
2. **File Generation:** These sequences are saved into files with a specified length.
3. **Compression:** The files are then compressed using modern compression algorithms.
4. **Length Comparison:** The length of the compressed file is compared against the critical value based on the alphabet size and original file length.
5. **Hypothesis Testing:** Finally, hypothesis testing is performed based on the comparison results.

### 2.2. NIST Test Suite

The NIST Test Suite is a battery of statistical tests developed by the National Institute of Standards and Technology (NIST) to assess the randomness of sequences [7].

Flow Chart Approach: The evaluation process using the NIST Test Suite typically follows these steps:

1. **Sequence Generation:** RNG sequences are generated.
2. **Test Application:** The generated sequences are subjected to various statistical tests provided by the NIST Test Suite.
3. **Result Analysis:** The results obtained from the tests are analyzed to determine the randomness and quality of the sequences.

### 2.3. TestU01 (Big-Crush) Statistical Test Suite

TestU01, particularly the Big-Crush suite, is a comprehensive set of statistical tests designed to evaluate the randomness and quality of random number generators. It includes a wide range of tests covering different aspects of randomness, such as uniformity, independence, and unpredictability [8].

Flow Chart Approach: The methodology utilizing the TestU01 suite generally involves the following steps:

1. **Random Number Generation:** RNG sequences are generated using selected algorithms.
2. **Test Suite Application:** The sequences are subjected to various statistical tests within the TestU01 suite, covering different aspects of randomness.
3. **Evaluation:** The results obtained from the tests are evaluated to determine the suitability and quality of the RNGs

## 3. Experimental Setup

The experimental setup involved comprehensive testing of RNGs using three distinct methodologies: NIST, TestU01, and data compression tests. The RNGs were evaluated across varying input sizes to ensure a thorough assessment of their performance. For the data compression tests, a range of compressors was employed, including Gzip [15], LZMA2[16], Brotli [18], Zstd [19], and LZFS [20]. These compressors were utilized to analyze the randomness of the generated sequences, with each RNG subjected to compression using different algorithms. Subsequently, the results obtained from the data compression tests, NIST tests, and L'Ecuyer-TESTU01 (BIG-CRUSH) tests were collected for analysis. This analysis aimed to evaluate the performance of each RNG under various testing conditions, including data compression, NIST, and L'Ecuyer-TESTU01 (BIGCRUSH) tests, providing insights into their effectiveness and reliability for generating random sequences.

### 3.1. Characteristics of Random number generators used in the experiment

The experiment involved the use of various Random number generators (RNGs), each with distinct characteristics. The Linear Congruential Generator [22], RANDU [23], LFSR [24], Middle Square Generator [25], Mersenne Twister data generator [26], PCG32[27], and Blum Blum Shub (B.B.S) [28] were employed. These RNGs varied in their methods and properties, ranging from simple equations to sophisticated algorithms. While some RNGs offered computational efficiency, others prioritized randomness or cryptographic security [17].

### 3.2. Characteristics of data compressors used in the experiment

It should be noted that data compression methods have been recognized in the literature as a foundation for randomness testing. For instance, the Lempel-Ziv Compression Test, and the Approximate Entropy Test are associated with universal codes, as discussed in [9]. Unlike traditional

methods, our approach offers the opportunity to conduct a randomness test based on any lossless data compression method, even when the distribution law of the code word lengths is unknown.

In our investigation of the randomness of Random number generators (RNGs), we utilized the following modern data compressors: Gzip [15], LZMA2 [16], Brotli [18], Zstd [19], and LZFSE [20]. The detailed characteristics of each compressor are shown below.

*Table 1*

Characteristics of Data compressor

| Data Compressors | Compression Ratio | Compression speed | Decompression Speed | Memory Usage | References |
|------------------|-------------------|-------------------|---------------------|--------------|------------|
| Gzip             | Moderate          | Fast              | Fast                | Low          | [15]       |
| LZMA2            | High              | Moderate to Slow  | Moderate to Fast    | High         | [16]       |
| Brotli           | Very High         | Moderate          | Moderate            | Moderate     | [18]       |
| Zstd             | Very High         | Fast              | Very Fast           | Moderate     | [19]       |
| LZFSE            | Medium            | Very Fast         | Very Fast           | Low          | [20]       |

### *3.2.1. Gzip*

Gzip, developed by Jean-loup Gailly and Mark Adler in 1992, is one of the most popular and widely supported compression algorithms. It utilizes the DEFLATE algorithm, which combines LZ77 compression and Huffman coding. Gzip offers moderate compression ratios and relatively fast compression and decompression speeds, making it suitable for various applications, including web servers, file compression, and transmission over networks. It is a standard tool in Unix-like systems and is commonly used to compress individual files or collections of files into a single archive [15].

### *3.2.2. LZMA2*

LZMA2 is a compression algorithm developed by Igor Pavlov for use in the 7-Zip archiver. It is based on the LZMA (Lempel-Ziv-Markov Chain-Algorithm) compression method, which combines LZ77-style sliding window compression with a range coder. LZMA2 offers high compression ratios and supports multi-threaded compression and decompression, enabling faster processing speeds on modern multi-core systems. It is often used in situations where maximum compression is desired, such as archiving files or distributing software packages. However, the trade-off for its high compression ratios is increased computational overhead during compression and decompression [16].

### *3.2.3. Brotli*

Developed by Google and released in 2015, Brotli is recognized for its exceptional compression ratios compared to many other algorithms. It was created to improve upon existing compression techniques by employing modern compression techniques like context modeling, Huffman coding, and a pre-defined dictionary. This approach allows Brotli to achieve compression ratios typically 20%-30% better than those of similar algorithms like Gzip and Deflate. Brotli also offers fast compression and decompression speeds, making it a popular choice for web servers and browsers to compress HTTP responses, reducing bandwidth usage and improving website loading times [18].

### 3.2.4. Zstd

Developed by Facebook and released in 2016, Zstandard (Zstd) is a compression algorithm designed for high performance and efficiency across a wide range of compression levels. It was created to strike a balance between compression ratio and speed, making it adaptable for various applications. Zstd can achieve compression ratios comparable to those of algorithms like Brotli while offering significantly faster compression and decompression speeds [19].

### 3.2.5. LZFSE

Developed by Apple and released in 2015, LZFSE (Lempel-Ziv Finite State Entropy) is optimized for use on iOS and macOS platforms. It was created by combining the Lempel-Ziv compression algorithm with finite state entropy coding to achieve high compression ratios while maintaining low memory and CPU usage. LZFSE was designed with efficiency in mind, particularly for devices with limited computational resources, such as smartphones and tablets. Its fast compression and decompression speeds, along with adaptive compression capabilities, make it an efficient choice for various data compression tasks [20].

## 4.0. Experimental Results

### 4.1. Linear Congruential Generator (LCG)

Table 2

LCG Data compression test result

| Compressor | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|------------|--------|--------|--------|----------------|--------|
| Gzip       | 100    | 100    | 100    | 100            | 100    |
| LZMA       | 100    | 100    | 100    | 100            | 100    |
| Brotli     | 100    | 100    | 100    | 100            | 100    |
| Zstd       | 100    | 100    | 100    | 100            | 100    |
| LZFSE      | 100    | 100    | 100    | 100            | 100    |

Table 3

LCG: NIST statically test results

| Test name                                   | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|---------------------------------------------|--------|--------|--------|----------------|--------|
| Frequency (Mono bit) Test                   | 0      | 0      | 0      | 0              | 0      |
| Frequency Test within a Block               | 0      | 0      | 0      | 0              | 0      |
| Runs Test                                   | 0      | 0      | 0      | 0              | 0      |
| Test for the Longest Run of Ones in a Block | 0      | 0      | 0      | 0              | 0      |
| Binary Matrix Rank Test                     | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform (Spectral) Test  | 100    | 100    | 100    | 100            | 100    |
| Non-overlapping Template Matching Test      | 0      | 0      | 0      | 0              | 0      |
| Overlapping Template Matching Test          | 0      | 0      | 0      | 0              | 0      |
| Maurer's 'Universal Statistical' Test       | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test                      | 0      | 0      | 0      | 0              | 0      |
| Serial Test                                 | 0      | 0      | 0      | 0              | 0      |
| Approximate Entropy Test                    | 0      | 0      | 0      | 0              | 0      |
| Cumulative Sums Test (Forward)              | 100    | 100    | 100    | 100            | 100    |
| Cumulative Sums Test (Backward)             | 100    | 100    | 100    | 100            | 100    |
| Random Excursions Test                      | 0      | 0      | 0      | 0              | 0      |
| Random Excursions Variant Test              | 0      | 0      | 0      | 0              | 0      |

Table 4

## LCG L'Ecuyer-TESTU01(BIG-CRUSH)

| Test Name                          | $10^3$ | $10^4$ | $10^5$ | $5*10^5$ | $10^6$ |
|------------------------------------|--------|--------|--------|----------|--------|
| Birthday Spacings Test             | 0      | 0      | 0      | 0        | 0      |
| Overlapping Permutations Test      | 0      | 0      | 0      | 0        | 0      |
| Collision Test                     | 100    | 100    | 100    | 100      | 100    |
| Close Pairs Test                   | 0      | 0      | 0      | 0        | 0      |
| Coupon Collector Test              | 100    | 100    | 100    | 100      | 100    |
| Gap Test                           | 100    | 100    | 100    | 100      | 100    |
| Matrix Rank Test                   | 0      | 0      | 0      | 0        | 0      |
| Linear Complexity Test             | 100    | 100    | 100    | 100      | 100    |
| Hamming Weight Test                | 0      | 0      | 0      | 0        | 0      |
| Hamming Correlation Test           | 0      | 0      | 0      | 0        | 0      |
| Hamming Independence Test          | 100    | 100    | 100    | 100      | 100    |
| String Test                        | 100    | 100    | 100    | 100      | 100    |
| Overlapping Template Matching Test | 100    | 100    | 100    | 100      | 100    |
| Quadrat Test                       | 100    | 100    | 100    | 100      | 100    |
| Permutation Test                   | 0      | 0      | 0      | 0        | 0      |
| Run Test                           | 0      | 0      | 0      | 0        | 0      |
| Poker Test                         | 100    | 100    | 100    | 100      | 100    |
| Poker Test (Poker 2)               | 0      | 0      | 0      | 0        | 0      |
| Gap Test (Gap 16K)                 | 100    | 100    | 100    | 100      | 100    |
| Gap Test (Gap)                     | 0      | 0      | 0      | 0        | 0      |
| Weighted Digits Test               | 100    | 100    | 100    | 100      | 100    |
| Weight Distribution Test           | 100    | 100    | 100    | 100      | 100    |
| Random Spheres Test                | 0      | 0      | 0      | 0        | 0      |
| Squeeze Test                       | 0      | 0      | 0      | 0        | 0      |
| Fourier Transform Test             | 0      | 0      | 0      | 0        | 0      |
| Discrete Fourier Transform Test    | 0      | 0      | 0      | 0        | 0      |

## 4.2. RANDU RNG

Table 5

## RANDU Data compression test result

| Compressor | $10^3$ | $10^4$ | $10^5$ | $5*10^5$ | $10^6$ |
|------------|--------|--------|--------|----------|--------|
| Gzip       | 0      | 100    | 100    | 100      | 100    |
| LZMA       | 0      | 55     | 100    | 100      | 100    |
| Brotli     | 0      | 100    | 100    | 100      | 100    |
| Zstd       | 0      | 0      | 1      | 100      | 100    |
| LZFSE      | 0      | 0      | 100    | 100      | 100    |

Table 6

## RANDU NIST test suit results

| Test name                                   | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|---------------------------------------------|--------|--------|--------|----------------|--------|
| Frequency (Monobit) Test                    | 0      | 0      | 0      | 0              | 0      |
| Frequency Test within a Block               | 0      | 0      | 0      | 0              | 0      |
| Runs Test                                   | 0      | 0      | 0      | 0              | 0      |
| Test for the Longest Run of Ones in a Block | 0      | 0      | 0      | 0              | 0      |
| Binary Matrix Rank Test                     | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform (Spectral) Test  | 100    | 100    | 100    | 100            | 100    |
| Non-overlapping Template Matching Test      | 0      | 0      | 0      | 0              | 0      |
| Overlapping Template Matching Test          | 0      | 0      | 0      | 0              | 0      |
| Maurer's 'Universal Statistical' Test       | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test                      | 0      | 0      | 0      | 0              | 0      |
| Serial Test                                 | 0      | 0      | 0      | 0              | 0      |
| Approximate Entropy Test                    | 0      | 0      | 0      | 0              | 0      |
| Cumulative Sums Test (Forward)              | 100    | 100    | 100    | 100            | 100    |
| Cumulative Sums Test (Backward)             | 100    | 100    | 100    | 100            | 100    |
| Random Excursions Test                      | 0      | 0      | 0      | 0              | 0      |
| Random Excursions Variant Test              | 0      | 0      | 0      | 0              | 0      |

Table 7

## RANDU L'Ecuyer-TESTU01(BIG-CRUSH)

| Test Name                          | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|------------------------------------|--------|--------|--------|----------------|--------|
| Birthday Spacings Test             | 0      | 0      | 0      | 0              | 0      |
| Overlapping Permutations Test      | 0      | 0      | 0      | 0              | 0      |
| Collision Test                     | 100    | 100    | 100    | 100            | 100    |
| Close Pairs Test                   | 0      | 0      | 0      | 0              | 0      |
| Coupon Collector Test              | 100    | 100    | 100    | 100            | 100    |
| Gap Test                           | 100    | 100    | 100    | 100            | 100    |
| Matrix Rank Test                   | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test             | 100    | 100    | 100    | 100            | 100    |
| Hamming Weight Test                | 0      | 0      | 0      | 0              | 0      |
| Hamming Correlation Test           | 0      | 0      | 0      | 0              | 0      |
| Hamming Independence Test          | 100    | 100    | 100    | 100            | 100    |
| String Test                        | 100    | 100    | 100    | 100            | 100    |
| Overlapping Template Matching Test | 100    | 100    | 100    | 100            | 100    |
| Quadrat Test                       | 100    | 100    | 100    | 100            | 100    |
| Permutation Test                   | 0      | 0      | 0      | 0              | 0      |
| Run Test                           | 0      | 0      | 0      | 0              | 0      |
| Poker Test                         | 100    | 100    | 100    | 100            | 100    |
| Poker Test (Poker 2)               | 0      | 0      | 0      | 0              | 0      |
| Gap Test (Gap 16K)                 | 100    | 100    | 100    | 100            | 100    |
| Gap Test (Gap)                     | 0      | 0      | 0      | 0              | 0      |
| Weighted Digits Test               | 100    | 100    | 100    | 100            | 100    |
| Weight Distribution Test           | 100    | 100    | 100    | 100            | 100    |
| Random Spheres Test                | 0      | 0      | 0      | 0              | 0      |
| Squeeze Test                       | 0      | 0      | 0      | 0              | 0      |
| Fourier Transform Test             | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform Test    | 0      | 0      | 0      | 0              | 0      |

### 4.3. Mersenne Twister data generator (MTRNG)

Table 8

MTRNG -Data compression test

| Compressor | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|------------|--------|--------|--------|----------------|--------|
| Gzip       | 0      | 0      | 0      | 0              | 0      |
| LZMA2      | 0      | 0      | 0      | 0              | 0      |
| Brotli     | 0      | 0      | 0      | 0              | 0      |
| Zstd       | 0      | 0      | 0      | 0              | 0      |
| LZFSE      | 0      | 0      | 0      | 0              | 0      |

Table 9

MTRNG NIST test suit results

| Test name                                   | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|---------------------------------------------|--------|--------|--------|----------------|--------|
| Frequency (Mono bit) Test                   | 0      | 0      | 0      | 0              | 0      |
| Frequency Test within a Block               | 0      | 0      | 0      | 0              | 0      |
| Runs Test                                   | 0      | 0      | 0      | 0              | 0      |
| Test for the Longest Run of Ones in a Block | 0      | 0      | 0      | 0              | 0      |
| Binary Matrix Rank Test                     | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform (Spectral) Test  | 100    | 100    | 100    | 100            | 100    |
| Non-overlapping Template Matching Test      | 0      | 0      | 0      | 0              | 0      |
| Overlapping Template Matching Test          | 0      | 0      | 0      | 0              | 0      |
| Maurer's 'Universal Statistical' Test       | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test                      | 0      | 0      | 0      | 0              | 0      |
| Serial Test                                 | 0      | 0      | 0      | 0              | 0      |
| Approximate Entropy Test                    | 0      | 0      | 0      | 0              | 0      |
| Cumulative Sums Test (Forward)              | 100    | 100    | 100    | 100            | 100    |
| Cumulative Sums Test (Backward)             | 100    | 100    | 100    | 100            | 100    |
| Random Excursions Test                      | 0      | 0      | 0      | 0              | 0      |
| Random Excursions Variant Test              | 0      | 0      | 0      | 0              | 0      |

Table 10

MTRNG L'Ecuyer-TESTU01(BIG-CRUSH)

| Test Name                     | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|-------------------------------|--------|--------|--------|----------------|--------|
| Birthday Spacings Test        | 0      | 0      | 0      | 0              | 0      |
| Overlapping Permutations Test | 0      | 0      | 0      | 0              | 0      |
| Collision Test                | 0      | 0      | 2      | 2              | 2      |
| Close Pairs Test              | 0      | 0      | 0      | 0              | 0      |
| Coupon Collector Test         | 0      | 0      | 2      | 2              | 2      |
| Gap Test                      | 100    | 100    | 100    | 100            | 100    |
| Matrix Rank Test              | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test        | 100    | 100    | 100    | 100            | 100    |
| Hamming Weight Test           | 1      | 1      | 2      | 2              | 2      |
| Hamming Correlation Test      | 0      | 0      | 0      | 0              | 0      |
| Hamming Independence Test     | 1      | 1      | 3      | 3              | 3      |
| String Test                   | 1      | 1      | 2      | 2              | 2      |

|                                    |     |     |     |     |     |
|------------------------------------|-----|-----|-----|-----|-----|
| Overlapping Template Matching Test | 100 | 100 | 100 | 100 | 100 |
| Quadrat Test                       | 100 | 100 | 100 | 100 | 100 |
| Permutation Test                   | 1   | 1   | 2   | 2   | 2   |
| Run Test                           | 1   | 1   | 2   | 2   | 2   |
| Poker Test                         | 1   | 1   | 1   | 1   | 1   |
| Poker Test (Poker 2)               | 0   | 0   | 1   | 1   | 1   |
| Gap Test (Gap 16K)                 | 100 | 100 | 100 | 100 | 100 |
| Gap Test (Gap)                     | 0   | 0   | 0   | 0   | 0   |
| Weighted Digits Test               | 100 | 100 | 100 | 100 | 100 |
| Weight Distribution Test           | 0   | 0   | 0   | 0   | 0   |
| Random Spheres Test                | 0   | 0   | 0   | 0   | 0   |
| Squeeze Test                       | 0   | 0   | 0   | 0   | 0   |
| Fourier Transform Test             | 0   | 0   | 0   | 0   | 0   |
| Discrete Fourier Transform Test    | 0   | 0   | 0   | 0   | 0   |

#### 4.4. PCG-32

Table 11

PCG Data compression test results

| Compressor | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|------------|--------|--------|--------|----------------|--------|
| Gzip       | 0      | 0      | 0      | 0              | 0      |
| LZMA       | 0      | 0      | 0      | 0              | 0      |
| Brotli     | 0      | 0      | 0      | 0              | 0      |
| Zstd       | 0      | 0      | 0      | 0              | 0      |
| LZFSE      | 0      | 0      | 0      | 0              | 0      |

Table 12

PCG NIST test suit result

| Test name                                   | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^5$ | $10^6$ |
|---------------------------------------------|--------|--------|--------|----------------|--------|
| Frequency (Mono bit) Test                   | 0      | 0      | 0      | 0              | 0      |
| Frequency Test within a Block               | 0      | 0      | 0      | 0              | 0      |
| Runs Test                                   | 0      | 0      | 0      | 0              | 0      |
| Test for the Longest Run of Ones in a Block | 0      | 0      | 0      | 0              | 0      |
| Binary Matrix Rank Test                     | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform (Spectral) Test  | 100    | 100    | 100    | 100            | 100    |
| Non-overlapping Template Matching Test      | 0      | 0      | 0      | 0              | 0      |
| Overlapping Template Matching Test          | 0      | 0      | 0      | 0              | 0      |
| Maurer's 'Universal Statistical' Test       | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test                      | 0      | 0      | 0      | 0              | 0      |
| Serial Test                                 | 0      | 0      | 0      | 0              | 0      |
| Approximate Entropy Test                    | 0      | 0      | 0      | 0              | 0      |
| Cumulative Sums Test (Forward)              | 0      | 0      | 0      | 0              | 0      |
| Cumulative Sums Test (Backward)             | 0      | 0      | 0      | 0              | 0      |
| Random Excursions Test                      | 0      | 0      | 0      | 0              | 0      |
| Random Excursions Variant Test              | 0      | 0      | 0      | 0              | 0      |

Table 13

PCG L'Ecuyer-TESTU01(BIG-CRUSH) result

| Test Name                          | $10^3$ | $10^4$ | $10^5$ | $5 \cdot 10^6$ | $10^6$ |
|------------------------------------|--------|--------|--------|----------------|--------|
| Birthday Spacing's Test            | 0      | 0      | 0      | 0              | 0      |
| Overlapping Permutations Test      | 0      | 0      | 0      | 0              | 0      |
| Collision Test                     | 0      | 0      | 2      | 2              | 2      |
| Close Pairs Test                   | 0      | 0      | 0      | 0              | 0      |
| Coupon Collector Test              | 0      | 0      | 2      | 2              | 2      |
| Gap Test                           | 0      | 0      | 0      | 0              | 0      |
| Matrix Rank Test                   | 0      | 0      | 0      | 0              | 0      |
| Linear Complexity Test             | 0      | 0      | 0      | 0              | 0      |
| Hamming Weight Test                | 2      | 2      | 0      | 0              | 0      |
| Hamming Correlation Test           | 0      | 0      | 0      | 0              | 0      |
| Hamming Independence Test          | 5      | 5      | 2      | 2              | 2      |
| String Test                        | 0      | 0      | 0      | 0              | 0      |
| Overlapping Template Matching Test | 0      | 0      | 0      | 0              | 0      |
| Quadrat Test                       | 0      | 0      | 0      | 0              | 0      |
| Permutation Test                   | 0      | 0      | 0      | 0              | 0      |
| Run Test                           | 0      | 0      | 0      | 0              | 0      |
| Poker Test                         | 2      | 2      | 2      | 2              | 2      |
| Poker Test (Poker 2)               | 1      | 1      | 0      | 0              | 0      |
| Gap Test (Gap 16K)                 | 0      | 0      | 0      | 0              | 0      |
| Gap Test (Gap)                     | 0      | 0      | 0      | 0              | 0      |
| Weighted Digits Test               | 0      | 0      | 0      | 0              | 0      |
| Weight Distribution Test           | 0      | 0      | 0      | 0              | 0      |
| Random Spheres Test                | 0      | 0      | 0      | 0              | 0      |
| Squeeze Test                       | 0      | 0      | 0      | 0              | 0      |
| Fourier Transform Test             | 0      | 0      | 0      | 0              | 0      |
| Discrete Fourier Transform Test    | 0      | 0      | 0      | 0              | 0      |

## 5.0. Discussion and Interpretation of the experimental results

We investigated the performance of four weak Random number generators (LCG, RANDU, LSFR, Middle Square Generator) and three strong random number generators (MT, PCG32 and Blum Blum shub) The evaluation was conducted using data compression tests, NIST, and L'Ecuyer-TESTU01 (BIG-CRUSH) methodologies across varying file sizes of 10KB, 100KB, 500KB and 1MB with each RNG subjected to 100 sequence repetitions.

The results of the experiment revealed that the weak generators, including LCG, RANDU RNG, LSFR, and Middle Square Generator, exhibited a 100% rejection rate in the data compression test, indicating poor randomness. Additionally, mixed results were observed in the NIST Test Suite, with significant rejections across various tests, highlighting weaknesses in the randomness of generated sequences. Similarly, the TestU01 (Big-Crush) analysis demonstrated significant rejections in multiple statistical tests, further indicating inadequacies in the randomness of the sequences generated by weak RNGs.

On the other hand, the strong generators, including MTRNG, PCG32, B.BS showed no significant rejection rates in the data compression test, suggesting better randomness compared to weak generators. Moreover, these strong generators exhibited strong performance in the NIST Test Suite, with a high passing rate across most tests, indicating robust randomness. Similarly, minimal rejections were observed in the TestU01 (Big-Crush) statistical tests for strong generators, further confirming their better performance and robust randomness compared to weak generators.

## 6.0. Conclusion

In conclusion, the effectiveness of the three tests applied to random number generators reveals important insights into their reliability and randomness. The data compression test, NIST Test Suite, and L'Ecuyer-TESTU01 (BIG-CRUSH) analysis serve as valuable tools for assessing the quality of RNG sequences.

The results indicate that data compressor test batteries such as Gzip, LZMA2, Brotli, Zstd, and LZSFSE effectively detect non-randomness in RNGs, while NIST and TestU01 do not.

The research results show that a test based on Modern data compressors has the potential to be as effective as other well-known test suites containing multiple tests.

These findings emphasize the importance of using multiple testing methodologies to assess RNG sequence randomness comprehensively and stress the potential of modern data compression tests as effective test suites.

## References

- [1] **Ryabko, B. Y.; Monarev, V. A.** Using information theory approach to randomness testing: Journal of Statistical Planning and Inference. 2005, vol. 133, no. 1, pp. 95–110.
- [2] **Ryabko, B.** Asymptotically most powerful tests for random number generators. J Stat. Planning Infer. 2022, 217, 1–7.
- [3] **L'ECUYER, P.** History of uniform random number generation. In 2017 Winter Simulation Conference. 2017, pp. 202–230.
- [4] **Mezenes, A.; Van Oorschot, P.; Vanstone, S.** Handbook of Applied Cryptography. CRC Press, Boca Raton, FL. 1996.
- [5] **Matsumoto, M.; Takuji, Nishimura.** Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. 8. 1998, 3-30.
- [6] **Rukhin, L.** Testing randomness: a suite of statistical procedures. Theory of Probability & Its Applications. 2001, vol. 45, no. 1, pp.111–132.
- [7] **Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; Dray, J.; Vo, S.** A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. National Institute of Standards and Technology, 2010.
- [8] **L'Ecuyer, P.; Simard, R.** TestU01: A C library for empirical testing of random number generators. ACM Trans. Math. Softw. (TOMS) 2007, 33, 22.
- [9] **Ryabko, B., Astola, J.; Malyutov, M.** Compression-Based Methods of Statistical Analysis and Prediction of Time Series. Springer International Publishing Switzerland. 2016.
- [10] **Marsaglia, G.** The marsaglia random number CDROM including the diehard battery of tests of randomness. 1995.
- [11] **Caelli, W.** Crypt x package documentation. Tech. Rep., Information Security Research. 1992.
- [12] **Katos, V.** A randomness test for block ciphers. Applied Mathematics and Computation. 2005, vol. 162, no. 1, pp. 29–35.

- [13] **Alcover, P. M., Guillam, A., Ruiz, M. D.** A new randomness test for bit sequences. *Informat-ics*. 2013, vol. 24, no. 3, pp. 339–356.
- [14] **Knuth, Donald. E.** The Art of Computer Programming. Semi numerical Algorithms. 1997, Volume 2, Addison-Wesley.
- [15] **Abdelfattah, M. S., Hagiescu, A., Singh, D.** Gzip on a chip: High performance lossless data compression on fpgas using opencl. In *Proceedings of the international workshop on openCL*. 2013 & 2014, pp. 1–9.
- [16] **Onuma, Y., Terashima, Y., Kiyohara, R.** Compression method for ECU software updates. IEEE Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU). 2017 Oct 3, pp. 1-6.
- [17] **Andrea, Rock.** Pseudorandom Number Generators for Cryptographic Applications. Universitat Salzburg, Marz. 2005.
- [18] **Alakuijala, J., Farruggia, A., Ferragina, P., Kliuchnikov, E., Obryk, R., Szabadka, Z., Vandevenne, L.** Brotli: *A general-purpose data compressor*. ACM Transactions on Information Systems (TOIS). 2018, 37, 1–30.
- [19] **Collet, Y., Kucherawy, M.** *Zstandard Compression and the application/zstd Media Type*. 2018, No. rfc8478.
- [20] **Li, Mingyin., Yue, Liu., Na, Wang.** A Novel ANS Coding with Low Computational Complexity *IEEE/CIC International Conference on Communications in China (ICCC Workshops)*. IEEE, 2023.
- [21] **Ryabko, B. Ya., Stognienko, V. S., Shokin, Yu. I.** A new test for randomness and its application to some cryptographic problems. *J. Statist. Plann. Inference* 123, 365–376.
- [22] **Fontaine, C.** LINEAR CONGRUENTIAL GENERATOR. van Tilborg, H.C.A. (eds) *Encyclopedia of Cryptography and Security*. Springer, Boston, MA. 2005.
- [23] **Markowsky, G.** The Sad History of Random Bits. *J. Cyber Secur. Mobil.* 2014, 31, 1–24.
- [24] **Klein, A.** Linear Feedback Shift Registers Stream Ciphers. Springer, London. 2013, 17–58.
- [25] **Kim, J., Chae, H.** All-Digital True Random-Number Generator using Middle Square Method, *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Taipei, Taiwan. 2022, pp. 1–3.
- [26] **Matsumoto, M., Saito, M., Nishimura, T., Hatagi, M.** Cryptographic Mersenne Twister and Fubuki stream/block cipher. Cryptographic ePrint Archive, June 2005.
- [27] **O'Neill, Melissa E.** PCG: A Family of Simple Fast Space-Efficient Statistically Good Algorithms for Random Number Generation. *ACM Transactions on Mathematical Software*. 2014
- [28] **Blum, L., Blum, M., Shub, M.** A Simple Unpredictable Pseudo-Random Number Generator. *SIAM Journal on Computing, Society for Industrial & Applied Mathematics (SIAM)*. 2021, 364–383.

### Information about the Author

Yeshewas Getachew Lulu, PhD Student

### Информация об авторе

Йешевас Гетачеу Лулу, аспирант

*Статья поступила в редакцию 30.01.2025;*

*одобрена после рецензирования 04.02.2025; принята к публикации 04.02.2025*

*The article was submitted 30.01.2025;*

*approved after reviewing 04.02.2025; accepted for publication 04.02.2025*

Научная статья

УДК 004.032.26

DOI 10.25205/1818-7900-2025-23-1-46-66

## Об одном подходе к разработке информационно-справочных систем на основе больших языковых моделей

Семен Евгеньевич Попов<sup>1</sup>, Вадим Петрович Потапов<sup>2</sup>  
Роман Юрьевич Замараев<sup>3</sup>

Федеральный исследовательский центр информационных и вычислительных технологий  
Новосибирск, Россия

<sup>1</sup>popov@ict.nsc.ru, <https://orcid.org/0000-0002-4763-2334>

<sup>2</sup>vadimptpv@gmail.com, <https://orcid.org/0000-0001-6959-3324>

<sup>3</sup>zamaraev@ict.nsc.ru, <https://orcid.org/0000-0003-4822-4794>

### Аннотация

В статье рассматривается современный подход к разработке контекстно-вопросно-ответной системы, направленной на обеспечение информационной поддержки процессов территориального управления. Основой системы являются большие языковые модели, использующие подход Retrieval-Augmented Generation (RAG). Источником данных для базы знаний послужили официальные доклады «О состоянии окружающей среды», включающие текстовую и табличную информацию. Для эффективного хранения и быстрого доступа к данным использовалась векторная база данных PostgreSQL с расширением pgvector. Основной компонент системы – RAG-модель, включающая Bi- и Cross-энкодеры, которые обеспечивают качественную обработку контекста и точный подбор релевантных ответов. Серверная часть системы разработана с использованием платформы Text Generation Interface, а клиентская часть реализована на базе ReactJS и Telegram-бота, что обеспечило удобство взаимодействия с пользователями. Результаты работы демонстрируют, что предложенный подход позволяет повысить точность и эффективность ответов за счет использования фильтров классификации запросов, адаптации временных интервалов и быстрого поиска релевантной информации. Разработанный графический интерфейс делает систему доступной для широкого круга пользователей. Полученные результаты подтверждают эффективность внедрения современных технологий искусственного интеллекта в задачи территориального управления. В перспективе предложенная система может быть адаптирована для других прикладных областей.

### Ключевые слова

большая языковая модель, контекстно-вопросно-ответная система, чат-бот, векторная база данных, RAG, Cross-Encoder, Bi-Encoder

### Финансирование

Исследование выполнено в рамках государственного задания Минобрнауки России для Федерального исследовательского центра информационных и вычислительных технологий.

### Для цитирования

Попов С. Е., Потапов В. П., Замараев Р. Ю. Об одном подходе к разработке информационно-справочных систем на основе больших языковых моделей // Вестник НГУ. Серия: Информационные технологии. 2025. Т. 23, № 1. С. 46–66. DOI 10.25205/1818-7900-2025-23-1-46-66

© Попов С. Е., Потапов В. П., Замараев Р. Ю., 2025

# On an Approach to Developing Information and Reference Systems Based on Large Language Models

Semyon E. Popov<sup>1</sup>, Vadim P. Potapov<sup>2</sup>  
Roman Y. Zamaraev<sup>3</sup>

Federal Research Center for Information and Computational Technologies  
Novosibirsk, Russian Federation

<sup>1</sup>popov@ict.nsc.ru, <https://orcid.org/0000-0002-4763-2334>

<sup>2</sup>vadimptpv@gmail.com, <https://orcid.org/0000-0001-6959-3324>

<sup>3</sup>zamaraev@ict.nsc.ru, <https://orcid.org/0000-0003-4822-4794>

## Abstract

The aim of this study is to develop a corporate context-aware question-answering system in the form of a chatbot to support territorial management by providing fast and accurate access to relevant information. The system is built upon large language models leveraging the Retrieval-Augmented Generation (RAG) approach, combined with modern data processing and retrieval techniques. The knowledge base of the system incorporates textual and tabular data extracted from official reports on environmental conditions. A PostgreSQL database with the pgvector extension was employed to store and retrieve large volumes of vectorized data efficiently. Advanced Bi- and Cross-encoder architectures were integrated to enhance the accuracy of context interpretation and the relevance of retrieved answers. The system's back-end was implemented using the Text Generation Interface platform, while the frontend relies on ReactJS and a Telegram bot, ensuring an intuitive user interface for diverse audiences. Additional features, such as query classification filters and temporal interval normalization, were introduced to refine the system's response precision and user experience. The results demonstrate the effectiveness of combining state-of-the-art artificial intelligence technologies with robust data management solutions. The developed system not only improves response accuracy and retrieval efficiency but also offers scalability and adaptability for application in various domains. This research highlights the potential of AI-driven solutions in addressing complex challenges in territorial management and beyond.

## Keywords

Large Language Model (LLM), Contextual Question-Answering System, Chatbot, Vector Database, Retrieval-Augmented Generation (RAG), Cross-Encoder, Bi-Encoder

## Funding

The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation for Federal Research Center for Information and Computational Technologies.

## For citation

Popov S. E., Potapov V. P., Zamaraev R. Y. On an Approach to Developing Information and Reference Systems Based on Large Language Models. *Vestnik NSU. Series: Information Technologies*, 2025, vol. 23, no. 1, pp. 46–66 (in Russ.) DOI 10.25205/1818-7900-2025-23-1-46-66

## Введение

Компьютерные информационно-справочные системы (ИСС), реализующие сбор, хранение, обработку и поиск текстовой, мультимедийной информации и данных составляют подавляющую часть программных решений в сети Интернет и в корпоративных сетях, и значительную часть на компьютерах и гаджетах конечных пользователей.

Основной функционал ИСС со стороны пользователя – это поисковая (под)система, состоящая из алгоритмов и программного обеспечения, предоставляющих возможность доступа к обширной коллекции как структурированной, так и неструктурированной информации (текст, изображение, звук) с помощью запросов [1].

Ключевым критерием качества работы поисковой системы являются релевантность – это степень соответствия документа запросу пользователя. С со стороны пользователя это означает, что в возвращенном поисковой системой списке потенциально полезных документов наиболее подходящий запросу документ должен находиться как можно выше, оптимально – первым.

В настоящее время классический, «документальный» подход к поиску известен, прежде всего, в публичных реализациях от IT-гигантов (Google, Microsoft, Yandex и др.) для работы с глобальной сетью Интернет. Однако создание и использование специализированных (корпоративных), частично или полностью изолированных (конфиденциальных) ИСС имеет ряд особенностей, в том числе необходимость получения обобщений и заключений для поддержки принятия решений.

Прежде всего для эффективного поиска документов необходимо, чтобы пользователь обладал достаточной компетенцией по теме запроса. Это позволяет сформулировать запрос лексически и синтаксически аналогично текстам в известных поисковой системе документах, что, в свою очередь, обеспечивает попадание релевантных документов в верхнюю часть списка в поисковом ответе. Также компетенция необходима, чтобы из нескольких документов со схожим содержанием выбрать наиболее адекватные задаче поиска. Кроме того, потребуются усилия, чтобы из многостраничного документа извлечь, в свою очередь, наиболее адекватные разделы и параграфы, и привести их к виду обобщения или заключения.

Эффективность поиска и обобщения в такой ситуации может быть повышена за счет использования:

- контекстов – уточняющих условий (даты, названия и т. п.), ключевых слов и словосочетания, вплоть до небольших фрагментов текста, сужающих область поиска;
- ожидаемых вопросов – подготовленных заранее наиболее популярных вопросов (Frequently Answered Question, FAQ) и готовых решений (How To) по важным темам, к которым подобраны наиболее вероятные вопросы, самые релевантные документы и сделаны обобщения.

Повсеместное внедрение таких решений ограничивалось чрезвычайной трудоемкостью предварительной обработки документов, особенно в части комплектования ожидаемых вопросов, ответов на них и формулировок готовых решений. В то же время как ответ на такую потребность в настоящее время наблюдается сверхбыстрое развитие технологий для работы с большими массивами документов. Эти технологии стали результатом впечатляющих достижений в области анализа данных и текстов на основе больших языковых моделей (БЯМ).

Принципиальным для темы статьи достижением в области БЯМ являются генеративные алгоритмы и сети на их основе [2], которые умеют «приходить» к ответу через цепочку «рассуждений» и «понимают» контекст, в том числе в виде медиаинформации (звук, изображение, видео). Здесь глаголы, используемые для описания рассудочной деятельности, сознательно взяты в кавычки, так как никакого интеллекта в этих алгоритмах, конечно же, нет.

Тем не менее современные генеративные сети (ChatGPT<sup>1</sup>, ее клоны и аналоги) [3] с определенного уровня сложности (размера) действительно способны создавать рефераты и производить обобщения текстов не хуже специалиста на основе общих принципов построения предложений и оформления логического заключения, заложенных в письменные языки.

Для выведения этих принципов требуется огромный объем и высокое качество обучающих текстов и сопровождающих их данных. Это неизбежно требует огромных трудозатрат на подготовку текстов и вычислительных мощностей для их анализа – обучения БЯМ.

Выигрыш для разработчика корпоративной ИСС при использовании БЯМ состоит в том, что ее можно использовать (скачать<sup>2</sup>, арендовать) в готовом, обученном виде, и с небольшими затратами доучить на корпоративных документах. При этом автоматически обеспечивается конфиденциальность и учет особенностей корпоративных данных.

Таким образом, актуальной видится задача разработки технологий создания корпоративных ИСС в виде чат-бота на основе генеративных БЯМ с обеспечением конфиденциальности и вычислительной доступности для широкого круга предприятий и организаций, и их сотрудников со средним уровнем компетенций в области анализа данных и БЯМ.

<sup>1</sup> <https://openai.com/index/chatgpt/>

<sup>2</sup> <https://huggingface.co/models>

## Постановки задачи

Разработка корпоративной контекстно-вопросно-ответной системы в виде чат-бота в рамках обеспечения информационной поддержки территориального управления.

## Исходные данные

Источником текстовой и табличной информации для базы знаний разработанной системы послужили официальные доклады «О состоянии окружающей среды» Министерства природных ресурсов и экологии Кемеровской и Новосибирской областей. Данные источники являются открытой информацией и расположены в сети Интернет<sup>3</sup>.

## Описание системы

Информационно-справочная система для поддержки принятия решений в задачах экологического мониторинга представляет собой контекстно-вопросно-ответную (КВО) систему, которая использует методы предобработки естественного языка (NLP) и интеграцию больших языковых моделей (LLM) с внешними источниками данных. Основной задачей КВО-системы является предоставление точной и релевантной информации пользователю в ответ на его запросы, учитывая контекст взаимодействия. КВО-система состоит из следующих ключевых элементов:

1. **Модуль обработки запросов.** Распознает и анализирует запрос пользователя. Проводит разбор естественного языка (NLU) для выделения сущностей, намерений и контекста.

2. **Модуль формирования ответов.** Генерирует релевантный и персонализированный ответ на основе запроса пользователя. Обеспечивает учет истории взаимодействий для создания контекстуальных ответов.

3. **Интеграция внешних источников данных.** Обеспечивает доступ к специализированным базам данных, отчетам, научным публикациям и другим источникам экологической информации. Использует поиск информации и систему извлечения знаний (IR и IE).

4. **Интерфейс взаимодействия с пользователем.** Обеспечивает удобное и интуитивно понятное взаимодействие пользователя с системой через текстовый или голосовой интерфейс.

Контекст в рамках КВО-системы – это текстовая информация, содержащая сведения, на основе которых модель должна сформулировать ответ. Вопрос – задание, сформулированное пользователем, на которое модель должна дать ответ. Вопросы могут быть прямыми или сложными, требующими анализа и логического вывода. Ответ – результат, который генерирует модель, интерпретируя контекст и используя информацию из него для ответа на обозначенный в вопрос.

В разработанной системе контексты представляют собой отдельные фрагменты текста, в основном один или несколько связанных по смыслу абзацев текста источника данных (см. раздел «Исходные данные»). Контексты формируют базу знаний КВО. Каждый контекст, по возможности, дополняется метаинформацией, включающей название раздела, подраздела и год отчета (рис. 1)

Для каждого годового доклада контексты сохраняются в соответствии с разделом в отдельный файл, например, для раздела «Климат» из доклада 2014 г. имя файла будет eco-report-2014-weather.pdf.

<sup>3</sup> <https://kuzbasseco.ru/doklady/o-sostoyanii-okruzhayushhej-sredy-kemerovskoj-oblasti/>; <https://mpr.nso.ru/page/3736>

```

{
  "section": "климат",
  "subsection": "погода, температура, осадки по месяцам",
  "year": "2014",
  "text": "В мае 2014 года наблюдалась неустойчивая погода с резкими колебаниями температуры и частыми обильными дождями во второй половине месяца. Наиболее теплой была первая декада мая, когда максимальная температура воздуха повышалась до +25, +27 °С. Минимальная температура воздуха понижалась до -1, -6 °С. Среднемесячная температура воздуха составила +7, +9 °С, что на 2 °С ниже нормы. По северным районам области осадков выпало 192-320 % от нормы, в остальных районах месячная сумма осадков составила 123-203 % от нормы."
}

```

Рис. 1. Фрагмент подготовленных контекстов базы знаний в json-формате  
 Fig. 1. A fragment of prepared knowledge base contexts in json format

Одним из ключевых элементов контекстно-вопросно-ответной системы является понятие «Промпт» (анг. prompt), а также механизмы его корректного формирования. Промпт используется для управления поведением модели, помогая ей понять задачу, ожидания пользователя и контекст, в рамках которого она должна работать. Контекст является основным строительным блоком промпт-инструкций больших языковых моделей. В рамках разработанной КВО используются промты-инструкции, которые позволяют отвечать на вопрос пользователя только на основе тех текстов базы знаний, к которым у КВО есть доступ.

### База знаний

Как указано в разделе «Исходные данные», база знаний формируется на основе докладов, представленных текстовыми файлами в формате docx и pdf.

Для более эффективного хранения контекстной информации, ее метаданных и эмбедингов была предложена концепция использования векторной базы. Векторные базы данных (Vector Databases) – это специализированные системы для хранения, поиска и работы с высокоразмерными эмбедингами, которые часто используются в системах Retrieval-Augmented Generation (RAG). Векторная база данных хранит информацию в виде числовых векторов, представляющих семантическое содержание текстов. Это позволяет эффективно искать документы или фрагменты текста, близкие к заданному запросу по смыслу, даже если ключевые слова между ними различаются. Для создания таких векторов обычно используют модели машинного обучения, такие как BERT, GPT и др. Эти модели обучаются преобразовывать тексты в векторы таким образом, чтобы похожие по смыслу тексты имели схожие векторы. Дополнительно могут храниться метаданные (например, текстовые или числовые атрибуты), которые связаны с каждым вектором для гибридного поиска.

Текстовая и векторная информация в векторной базе данных индексируется. **Индексирование** – это процесс подготовки данных для быстрого поиска ближайших соседей (Approximate Nearest Neighbor Search, ANNs). Выделяют следующие методы индексации: Flat Brute-Force, HNSW (Hierarchical Navigable Small World Graph), IVF (Inverted File Index), LSH (Locality-Sensitive Hashing).

На текущий момент существует большое количество специализированных систем управления базами данных, поддерживающих хранение и обработку векторных представлений для текстовой информации. Ниже приведена таблица с некоторыми из них, с их достоинствами и недостатками (табл. 1.)

Таблица 1

Достоинства и недостатки некоторых векторных баз данных

Table 1

Advantages and disadvantages of some vector databases

| Название                                                  | Достоинства                                                                                | Недостатки                                                                                       | Области применения                                               |
|-----------------------------------------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| <b>Milvus<sup>4</sup></b>                                 | Поддержка больших объемов данных; высокая производительность; интеграция с ML-фреймворками | Требуется настройка инфраструктуры; высокие требования к ресурсам при работе с большими данными  | Поиск по изображениям, видео, текстам; рекомендательные системы  |
| <b>Pinecone<sup>5</sup></b>                               | Облачное решение; простота в использовании; автоматическое масштабирование                 | Зависимость от облака; платное решение                                                           | Быстрое прототипирование; поиск похожих данных в веб-приложениях |
| <b>FAISS<sup>6</sup></b>                                  | Высокая производительность; оптимизация под GPU; поддержка пользовательских индексов       | Не поддерживает гибридный поиск; отсутствие встроенных механизмов масштабирования                | Локальные проекты; эксперименты с большими наборами данных       |
| <b>ElasticSearch<sup>7</sup> / OpenSearch<sup>8</sup></b> | Гибридный поиск (векторы + текст); хорошо известный стек технологий                        | Ограниченная оптимизация для векторов; трудности с настройкой масштабирования                    | Универсальные поисковые системы                                  |
| <b>pgvector<sup>9</sup></b>                               | Интеграция с PostgreSQL; простота настройки; подходит для гибридного поиска                | Ограниченная производительность на больших объемах; меньшая оптимизация ANNs по сравнению с HNSW | Небольшие RAG-системы; гибридный поиск                           |

Выбор базы данных зависит от конкретных требований объема данных, доступных ресурсов и ожидаемой производительности. Для разработанной системы, принимая во внимание нижеперечисленные преимущества, была выбрана СУБД PostgreSQL с расширением pgvector.

1. Унифицированное хранилище данных позволило использовать единую базу данных PostgreSQL как для регулярных данных системы, так и для векторных данных.

2. Расширенные SQL-возможности СУБД имплементируют комбинированный поиск по векторным данным с традиционными SQL-запросами, включая фильтры, агрегации, соединения, пространственные данные и прочее. Возможность хранить векторы вместе с метаданной (текстовые данные, числовые атрибуты), что упрощает выполнение гибридных запросов.

3. Удалось снизить до минимума зависимости в проекте, так как расширение pgvector не требует установки сложных инструментов, как в случае с отдельными векторными базами. Также позволило использовать настроенными системами мониторинга и резервного копирования корпоративного кластера СУБД.

<sup>4</sup> <https://milvus.io/>

<sup>5</sup> <https://www.pinecone.io/>

<sup>6</sup> <https://github.com/facebookresearch/faiss>

<sup>7</sup> <https://www.elastic.co/elasticsearch>

<sup>8</sup> <https://opensearch.org/>

<sup>9</sup> <https://github.com/pgvector/pgvector>

4. СУБД PostgreSQL, развернутая в кластерном исполнении, практически ни в чем не уступает специализированным решениям в задачах по хранению, извлечению и обслуживанию векторных представлений текстовых данных, например таких, как Elasticsearch или OpenSearch.

5. Расширение pgvector имеет богатые возможности для работы с векторными данными: методы поиска на основе алгоритмов ближайших соседей, таких как L2 (евклидово расстояние), косинусное расстояние и расстояние Минковского; создание индексов для ускорения поиска (например, HNSW); поддержка высокоразмерных векторов, что удобно для работы с эмбедингами создаваемыми современными большими языковыми моделями.

6. Легкость интеграции. Поддержка языка программирования Python через библиотеки, такие как SQLAlchemy, Psycopg и другие ORM.

Таким образом, выбор программного решения pgvector для работы с базой знаний контекстно-вопросно-ответной системы видится вполне актуальным благодаря гибкости, масштабируемости и интеграции в СУБД PostgreSQL, а также расширенным возможностям оперирования векторными данными.

Для хранения эмбедингов была создана таблица «contexts» со следующими столбцами: id – номер записи, тип: index; text – исходный текст, тип: text; year – год доклада, тип: varchar(4); section – название раздела в докладе, тип: varchar(128); subsection – название подраздела в разделах доклада, тип: varchar(256); embeddings – столбец, содержащий массивы эмбедингов, тип: vector(1024).

Для создания эмбедингов использовалась модель d0rj/e5-large-en-ru<sup>10</sup>. При этом предварительно подготовленные контексты в виде файлов json-формата (рис. 1) загружались в векторную базу данных в два этапа. На первом этапе данные кодируются в эмбединги, используя энкодер модели d0rj/e5-large-en-ru и составляется SQL-инструкция с дополнительной метainформацией для каждого из них. На втором этапе подготовленные инструкции выполняются в виде транзакции вставки построчно в базу данных (рис. 2). Таким образом получаем заполненную базу данных (знаний), где каждая запись несет полную информацию для успешного извлечения релевантных текстов и формирования промпт-инструкции согласно запросу пользователя (табл. 2).

Таблица 2

Пример записи в базе данных для представления контекстов в виде эмбедингов

Table 2

Example of a database entry for representing contexts as embeddings

| id   | text                                                                                                                                                                                                                                                                                                                                                                                | years | section | subsection                             | embeddings                                                |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|---------|----------------------------------------|-----------------------------------------------------------|
| 2054 | В январе 2012 г. наблюдалась неустойчивая, со значительным недобором осадков погода. Максимальная температура воздуха повышалась до –10...–15 °С. Минимальная температура воздуха понижалась до –36...–40 °С. Среднемесячная температура воздуха в январе составила –20...–23 °С, что на 3–6 °С ниже нормы. Осадки наблюдались преимущественно слабые и составили 22–65 % от нормы. | 2012  | климат  | погода, температура, осадки по месяцам | [1,1712, –0,0270, –1,6838, ..., 0,6163, –0,7662, –0.0960] |

<sup>10</sup> <https://huggingface.co/d0rj/e5-large-en-ru>

```

...
model = SentenceTransformer("d0rj/e5-large-en-ru")
model.to("cuda")
...
input_files = [eco-report-2014-weather.pdf.json,...]

# Инициализация соединения с БД
connection = psycopg2.connect(database="...", user="...", password="...",
                             host="...", port="...")
# Создание курсора соединения
cursor = connection.cursor()

# Каждый файл контекстов открываем в json-формате
for input_file in input_files:
    file = open(input_file, encoding="utf-8")
    data = json.load(file)
    ....

    # Читаем и обрабатываем каждую json-запись
    for context in data:
        # Читаем поля текст, год, раздел, подраздел
        text = context["text"]
        year = context["year"]
        section = context["section"]
        subsection = context["subsection"]

        # Создаем эмбединги
        text_embeddings = model.encode(text, show_progress_bar=False,
                                       convert_to_numpy=True,
                                       normalize_embeddings=True)
        # Ковертируем массив Python в формат SQL-массива.
        embeddings_pg_vector = f"['{','.join(map(str, text_embeddings))}']"

        # Формируем полную SQL-инструкцию.
        sql = (f"INSERT INTO rag.contexts (text, years, months, section,
                                         subsection, keywords, embeddings) "
              f"VALUES ('{text}', '{years}', '{months}', '{section}', "
              f"'{subsection}', '{keywords}', {embeddings_pg_vector})")
        # Добавляем инструкцию в массив на исполнение
        sqls.append(sql)

# Вставляем данные в таблицу contexts, используя механизм транзакций
for idx, sql in sqls:
    cursor.execute(sql)
connection.commit()
....

```

Рис. 2. Фрагмент кода на языке Python для загрузки текстов в векторную базу

Fig. 2. A fragment of Python code for uploading texts to a vector database

## RAG-система

Одной из ключевых проблем организации функционирования КВО является эффективность, точность и автоматизация построения промптов, релевантных запросу пользователя. Для решения данной проблемы был использован подход Retrieval-Augmented Generation (RAG). RAG – это технология, сочетающая в себе поиск релевантной информации в существующих хранилищах данных и генерацию текста с помощью языковых моделей, для создания более точных и информативных ответов [4]. Использование RAG в разработанной КВО-сис-

теме позволило обойти ряд ограничений, присущих таким специализированным системам, работающих на малых графических кластерах (не более 8 GPU):

- **Ограничение на количество токенов.** Современные большие языковые модели имеют ограничение на максимальное количество токенов, которое они могут обрабатывать за раз. Это делает невозможным хранение и обработку большого объема информации непосредственно внутри модели.
- **Актуальность данных.** Языковые модели обучаются на статических данных, которые не совсем актуальны для, например, региональной экологии. RAG позволяет получать актуальную информацию из внешних источников (векторных баз данных) в режиме реального времени.
- **Экономия ресурсов.** Использование RAG снижает нагрузку на языковую модель, так как большая часть знаний находится во внешних источниках и загружается только при необходимости. RAG состоит из двух основных компонентов: модуль извлечения (Retriever) и модуль генерации (Generator). Модуль извлечения ищет релевантную информацию в подключенной векторной базе данных (например, PostgreSQL/pgvector). Модуль генерации (Generator) создает текстовые ответы на основе данных, предоставленных модулем извлечения.

В разработанной системе для поиска и извлечения релевантной информации используются подходы, основанные на применении менее сложных и малых по параметрам нейронных сетей, а именно Bi-Encoder и Cross-Encoder. Bi-Encoder и Cross-Encoder являются подходами, используемыми в контексте плотного поиска (dense retrieval).

Bi-Encoder – это подход, который строит отдельные векторные представления для запроса и документа (контекста). Преимущества Bi-Encoder включают высокую скорость обработки, эффективность использования памяти, масштабируемость и простоту интеграции. Так как запрос и документ кодируются отдельно двумя экземплярами одной и той же модели, то векторы запросов и документов (текстов) проецируются в общее пространство, где сходство между ними вычисляется с помощью метрик, таких как косинусное расстояние или скалярное произведение. Такой подход позволяет формировать эмбединги только для запроса пользователя, а вычислять метрику на основе запроса средствами векторной СУБД. Основываясь на этих преимуществах, Bi-Encoder позволяет достаточно быстро сформировать пул из, например, 30–80 корпусов наиболее релевантных текстов для последующей передачи их в более медленный, но наиболее точный Cross-Encoder. На рис. 3 представлен код извлечения текстов из векторной базы данных в модуле Bi-Encoder.

Bi-Encoder имеет некоторые ограничения в плане точности и релевантности между запросом и извлекаемыми текстами – это отсутствие прямого взаимодействия на этапе кодирования, что может приводить к потере информации о контексте. Другими словами, релевантные тексты могут получить меньшее значение, например, евклидова расстояния. Следовательно, для гарантированного «охвата» нужных, необходимо оставлять большее их количество в результате запроса. Для нивелирования данного ограничения был использован подход на основе Cross-Encoder.

Cross-Encoder использует модель для совместного кодирования запроса и документа, что позволяет учитывать их взаимодействие на этапе кодирования. Запрос и документ конкатенируются и подаются в один общий трансформер. Модель генерирует представление для их комбинации и вычисляет оценку релевантности (например, вероятность, что документ соответствует запросу). Поэтому точность ранжирования и, следовательно, оценка релевантности у подхода с использованием Cross-Encoder заметно выше, чем у Bi-Encoder.

В текущей версии КВО для подхода Cross-Encoder была использована дообученная модель all-MiniLM-L6-v2<sup>11</sup>. Дообучение проводилось на специальной выборке MMarco<sup>12</sup> для русского языка. Выборка состояла из обучающих и валидационных записей (рис. 4, 5).

<sup>11</sup> <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

<sup>12</sup> <https://huggingface.co/datasets/unicamp-dl/mmarco/tree/main>

```

...
model = SentenceTransformer('d0rj/e5-large-en-ru', device="cuda")
...
# Получаем массив эмбедингов для пользовательского
# запроса (переменная question)
embeddings = model.encode(question, convert_to_numpy=True,
                           normalize_embeddings=True)
# Ковертируем массив Python в формат SQL-массива.
embeddings_pg_vector = f"[{' '.join(map(str, embeddings))}]"
....
# Выполняем SQL-инструкцию по поиску релевантных
# тестов на основе евклидовых расстояний
# для более точного ответа в SQL-инструкцию
# добавляется метаданная (год, раздел, подраздел)
query_result_rows = cursor.execute("SELECT text,
    inner_product(embeddings, %s) as inner_product
    FROM rag.contexts
    WHERE year=%s AND subsection = %s
    ORDER BY inner_product DESC LIMIT %s",
    (embeddings_pg_vector, years, subsection,
    context_fetch_limit)
    )

# Создаем массив текстов для последующей передачи
# в Cross-Encoder
texts = []
for row in query_result_rows:
    texts.append(row[0])

```

Рис. 3. Фрагмент кода извлечения релевантных текстов на основе подхода Bi-encoder и векторной базы данных PostgreSQL/pgvector

Fig. 3. A fragment of code for retrieving relevant texts based on the Bi-encoder approach and PostgreSQL/pgvector vector database

```

{
  "text": "В 2021 году в Новокузнецком городском округе максимальная из разовых  
концентрация оксида углерода 2,7 ПДК и наибольшая повторяемость проб выше ПДК - 2,0 %  
отмечены в Центральном районе.",
  "question": "В Центральном районе была определена максимальная концентрация оксида  
углерода за один раз - какая она была?",
  "rank": 4.4
}

```

Рис. 4. Пример элемента обучающей выборки для модели all-MiniLM-L6-v2

Fig. 4. Example of a training sample for the all-MiniLM-L6-v2 model

```

{
  "query": "Что такое валовая заработная плата, включая надбавку?",
  "positive": [
    "Компоненты валовой заработной платы включают базовую заработную плату, надбавку за дороговизну, надбавку за аренду дома, городскую компенсационную надбавку и другие вознаграждения."
  ],
  "negative": [
    "Выигрыши в азартных играх. Денежные выигрыши включаются в федеральный валовой доход и, следовательно, должны быть включены в валовой доход штата Массачусетс. Помимо денежных выигрышей, выигрыши в азартных играх также включают справедливую рыночную стоимость безналичных призов, таких как автомобили, дома и поездки.",
    "Для новоприбывших около 15 процентов ежемесячной брутто-зарплаты работника вычитаются работодателем из части налога на заработную плату правительства Бермудских островов.",
    ....
  ]
}

```

Рис. 5. Пример элемента валидационной выборки для модели all-MiniLM-L6-v2

Fig. 5. Example of a validation sample for the all-MiniLM-L6-v2 model

Данная большая выборка является аккумулярованием различных датасетов, составленных различными группами исследователей. Ранги проставлялись вручную, равно как и формировались наборы «positive» и «negative» в валидационных датасетах. Общий размер выборки порядка 9 Гб текстовой информации. Общее время дообучения модели составило около 5 дней на шести видеокартах Nvidia A100 (80 Гб VRAM).

Отличие обучающих и валидационных выборок обусловлено требуемой спецификой работы подхода Cross-Encoder. А именно, было важным не только назначить ранги соответствия запрос – текст, но и получить правильный порядок вывода текстов из базы данных. Применение простой сортировки в данном случае было не совсем уместно. Модель должна во время обучения сопоставлять вычисленный ранг с местом (порядком) следования текстов. Данный результат был достигнут применением комбинации двух метрик: MRR (Mean Reciprocal Rank) и NDCG (Normalized Discounted Cumulative Gain)<sup>13</sup>.

Наряду с преимуществами подход Cross-Encoder также имеет свои ограничения. Это медленное выполнение ввиду невозможности предобработать тексты, т. е. каждая пара запрос – текст должна быть вычислена в реальном времени. Следовательно, для увеличения точности выбора релевантных текстов относительно пользовательского запроса необходимо использовать комбинацию Bi-Encoder и Cross-Encoder. То есть сформировать пул из, например, 30–50 корпусов текстов при помощи подхода Bi-Encoder и передать их на вход в более медленный, но наиболее точный Cross-Encoder, и уже на основе его классификатора проводить переранжирование по степени схожести, в результате оставляя 2–3 их них, из которых потом формировать окончательный промпт для большой языковой модели.

Для еще более точного извлечения релевантных текстов из базы данных применялись дополнительные фильтры перед отправкой пользовательского запроса в модуль Bi-Encoder. Фильтры включали проверку орфографии, классификацию вопроса по подразделам корпусов

<sup>13</sup> <https://blog.stackademic.com/ndcg-vs-mrr-ranking-metrics-for-information-retrieval-in-rags-2061b04298a6>

текстов, а также фильтрацию запроса (по необходимости) по годам и месяцам для сужения области поиска и уточнения контекстной информации для LLM.

Для проверки орфографии использовались сервис проверки орфографии от Яндекс<sup>14</sup>, а также в случае его недоступности модель ai-forever/RuM2M100-1.2B<sup>15</sup>. Классификация вопроса по подразделам корпусов текстов проводилась с помощью специально дообученной модели google/mt5-large<sup>16</sup>. Дообучение проводилось на подготовленном датасете, включающим различные варианты запросов пользователя и их соответствия подразделам базы знаний (рис. 6). Всего было подготовлено более 17 000 вопросов по 20 разделам.

```
{
  "question": "Какой процентный рост массы выбросов загрязняющих веществ
    произошел в Калтанском городском округе в 2020 году по сравнению с
    предыдущим годом?",
  "class_id": 12,
  "class_name": "выбросы загрязняющих веществ по административным террито"
```

Рис. 6. Пример элемента обучающей выборки для классификации запроса пользователя в название подраздела

Fig. 6. Example of a training sample for classifying a user query into a subsection title

Для определения года, месяца, замены номеров месяцев, кварталов, различных выражений, обозначающих временные промежутки и т. п., использовались нативные регулярные выражения языка Python. Целью их использования была замена нерелевантных фрагментов временных выражений в запросе пользователя на использование только названий месяцев и цифровых обозначений годов. Например, для грамматически неверного вопроса «Какая *срдня температура* была во втором квартал 2014 года в Кемеровской области?», последовательное применение дополнительных фильтров вернет преобразованный вопрос «Какая средняя температура была в апреле, мае, июне 2014 года в Кемеровской области?», определит, что он относится к докладам 2014 года и подразделу «Погода, температура, осадки по месяцам», при этом работает шаблон регулярных выражений следующего вида «(\b[авдкнопс]\*\b)\s\*(\b[авгдziejклмнопрстцщшщюя]\*\b)\s\*(1|2|3|4)\s\*квартал\w{0,3}».

Тестирование вышеуказанных подходов показало, что точность извлечения релевантных корпусов текстов повысилась с 50 до 93,7 %. А именно, частота попадания релевантного корпуса в набор повысилась до 99 % против 73 % (только Bi-Encoder с косинусным расстоянием), при этом релевантный контекст всегда попадал в 5 лучших текстов, отсортированных по убыванию ранга. Тестирование проводилось на 567 различных тематических пользовательских вопросах, подсчет и проверка осуществлялась вручную с поиском правильного ответа непосредственно в тексте файлов официальных докладов «О состоянии окружающей среды».

Ниже представлен полный вид промпт-инструкции для большой языковой модели КВО (рис. 7).

Представленная версия КВО поддерживает ответы на запросы пользователя в табличном виде. Таблицы в тексте докладов были переведены в таблицы базы данных PostgreSQL. Текущая версия системы может работать с таблицами, представленными в первой форме нормализации и не имеющими соединений по вторичному ключу. Для корректного составления промп-

<sup>14</sup> <https://speller.yandex.net/>

<sup>15</sup> <https://huggingface.co/ai-forever/RuM2M100-1.2B>

<sup>16</sup> <https://huggingface.co/google/mt5-large>

та для ответа в табличном виде использовался дополнительный модуль RAG-системы. Данные модуля позволяют классифицировать вопрос пользователя в название таблицы. Для классификации применялась дообученная модель google/mt5-large. Дообучение проводилось на подготовленном датасете, включающем различные варианты запросов пользователя и их соответствия названиям таблиц в базе данных PostgreSQL (рис. 8). Всего было подготовлено более 6 245 вопросов для 6 таблиц.

```
{
  "messages": [
    {
      "role": "system",
      "content": "Ты - ИИ-помощник. Веди себя так, как будто у тебя нет других источников информации кроме Текста. Внимательно следуй тексту. Ты должен дать подробный ответ на Вопрос, используя данный Текст."
    }, {
      "role": "user",
      "content": "Текст:\nВ июле 2019 года умеренно прохладная, с недобором осадков, дымкой, утренними туманами погода. Максимальная температура воздуха была от +25 до +32 °С. Минимальная температура воздуха понижалась до +7, +10 °С. Среднемесячная температура воздуха составила +17, +19 °С, что на 1 °С ниже нормы. Ливневые дожди с грозами наблюдались по большинству районов. В сумме за месяц осадков выпало 40-83 % нормы."
    }, {
      "role": "bot",
      "content": "В июле 2019 года в области была умеренно прохладная погода с недобором осадков, дымкой, утренними туманами. Максимальная температура воздуха варьировалась от +25 до +32 °С, а минимальная температура понижалась до +7, +10 °С. Среднемесячная температура составила +17, +19 °С, что на 1 °С ниже нормы. Ливневые дожди с грозами были характерны по большинству районов, а суммарное количество осадков за месяц составило 40-83 % от нормы."
    }
  ]
}
```

Рис. 7. Промпт-инструкция для вопроса/ответа БЯМ в текстовом виде. Элемент массива «messages» с ключом «role», равное «bot», добавляется при ответе БЯМ на запрос пользователя

Fig. 7. Prompt instruction for question/answer LLM in textual format. The “messages” array element with the key “role” equal to “bot” is added when the LLM responds to a user query

```
{
  "question": "В каком году в городе Новокузнецке было наибольшее количество повторений северного направления ветра?",
  "class_id": 5,
  "class_name": "repeat_wind_direction_calms"
}
```

Рис. 8. Пример элемента обучающей выборки для классификации запроса пользователя в название таблицы

Fig. 8. Example of a training sample for classifying a user query into a table title

```

SELECT 'CREATE TABLE ' || pc.relname || E'(\n' ||
    string_agg(pa.attname || ' ' || pg_catalog.format_type(pa.atttypid,
        pa.atttypmod) || coalesce(' DEFAULT ' || (
        SELECT pg_catalog.pg_get_expr(d.adbin, d.adrelid)
        FROM pg_catalog.pg_attrdef d
        WHERE d.adrelid = pa.attrelid
        AND d.adnum = pa.attnum
        AND pa.atthasdef
    ), '')) || ' ' ||
    CASE pa.attnotnull
        WHEN TRUE THEN 'NOT NULL'
        ELSE 'NULL'
    END, E',\n') ||
    coalesce((SELECT E',\n' || string_agg('CONSTRAINT ' ||
        pc1.conname || ' ' ||
        pg_get_constraintdef(pc1.oid), E',
        \n' ORDER BY pc1.conindid)
        FROM pg_constraint pc1
        WHERE pc1.conrelid = pa.attrelid), '') ||
    E');'
FROM pg_catalog.pg_attribute pa
    JOIN pg_catalog.pg_class pc
        ON pc.oid = pa.attrelid
        AND pc.relname = 'repeat_wind_direction_calms'
    JOIN pg_catalog.pg_namespace pn
        ON pn.oid = pc.relnamespace
        AND pn.nspname = 'rag_tables'
WHERE pa.attnum > 0
    AND NOT pa.attisdropped
GROUP BY pn.nspname, pc.relname, pa.attrelid;

```

Рис. 9. SQL-инструкция для генерации DDL-инструкции создания выбранной таблицы (таблица «repeat\_wind\_direction\_calms», схема «rag\_tables»)

Fig. 9. SQL instruction for generating the DDL statement to create the selected table (table “repeat\_wind\_direction\_calms”, schema “rag\_tables”)

После классификации вопроса, на основе имени таблицы, используя специальный SQL-запрос (рис. 9), формировалась ее DDL-инструкция вместе с описанием столбцов и таблицы. Данная DDL-инструкция включалась в промпт-инструкцию для вопроса/ответа в табличном виде (рис. 10).

### Серверная часть системы

В текущей версии в ядре разработанной системы поддерживаются следующие большие языковые модели: casperhansen/llama-3-70b-fp16<sup>17</sup> для ответов в текстовом виде и defog/sqlcoder-70b<sup>18</sup> для ответов в табличном виде. Для ускорения ответов БЯМ была использована технология Text Generation Interface (TGI)<sup>19</sup>. Запуск моделей производился параллельно на не-

<sup>17</sup> <https://huggingface.co/casperhansen/llama-3-70b-fp16>

<sup>18</sup> defog/sqlcoder-70b-alpha

<sup>19</sup> <https://github.com/huggingface/text-generation-inference?tab=readme-ov-file#run-locally>

```

{
  "messages": [
    {
      "role": "user",
      "content": "## Task\n\tGenerate a SQL query to answer \n\t[QUESTION]Покажи  

      все информацию из таблицы \"Средняя температура  

      воздуха\"?[/QUESTION]\n\n\t### Instructions\n\t- If you cannot answer the  

      question with the available database schema, return 'I do not know'\n\n\t###  

      Database Schema\n\tThe query will run on a database with the following  

      schema:\n\tCREATE TABLE average_monthly_air_temperature(id integer NOT NULL,  

      month character varying(1024) NULL,\n\tvalue numeric(5,1) NULL, year  

      integer NULL, locality character varying(1024) NULL,\n\tCONSTRAINT  

      average_monthly_air_temperature_pkey PRIMARY KEY (id));\n\n\tCOMMENT ON  

      TABLE average_monthly_air_temperature IS 'Средняя температура воздуха,  

      С';\n\tCOMMENT ON COLUMN id IS NULL;\n\tCOMMENT ON COLUMN month IS  

      'Месяц';\n\tCOMMENT ON COLUMN value IS 'Средняя температура';\n\tCOMMENT ON  

      COLUMN year IS 'Год';\n\tCOMMENT ON COLUMN locality IS 'Административная  

      территория';\n\n\t### The average_monthly_air_temperature table Data  

      Example:\n\tThe COLUMN 'id' can contain any numeric data in range from 1 to  

      72\n\tThe COLUMN 'month' can contain any text data such as: Ноябрь, Апрель,  

      Декабрь, Июнь ... \n\tThe COLUMN 'value' can contain any numeric data in  

      range from -20.7 to 19.6\n\tThe COLUMN 'year' can contain any numeric data  

      in range from 2020 to 2022\n\tThe COLUMN 'locality' can contain any text  

      data such as: Новокузнецк, Кемерово"
    },
    {
      "role": "bot",
      "content": "### Answer\n\tGiven the database schema, here is the SQL query  

      that answers \n\t[QUESTION]Покажи все информацию из таблицы \"Средняя  

      температура воздуха\"?[/QUESTION]\n\t[SQL]\n\t'SELECT * FROM  

      average_monthly_air_temperature;'"
    }
  ]
}

```

Рис. 10. Промпт-инструкция для вопроса/ответа БЯМ в табличном виде. Элемент массива «messages» с ключом «role», равное «bot», добавляется при ответе БЯМ на запрос пользователя

Fig. 10. Prompt instruction for question/answer LLM in tabular format. The “messages” array element with the key “role” equal to “bot” is added when the LLM responds to a user query

скольких GPU с использованием Text Generation Interface (TGI). Это позволило повысить производительность и скорость генерации текста. Основные механизмы параллельного запуска БЯМ включали:

- модульную декомпозицию (Model Sharding). Модель разбивается на части (шарды), и каждая часть размещается на отдельном GPU. Это полезно для размещения больших моделей, которые не помещаются в память одного устройства;

- параллельное выполнение (Tensor Parallelism). Вычисления для одной модели распределяются между несколькими GPU. Используется для ускорения обработки запросов;
- обработка запросов (Batching). Пакетная обработка запросов распределяется между GPU. Это позволяет эффективно использовать ресурсы;
- оптимизация передачи данных. Синхронизация между GPU минимизируется за счет использования оптимизированных библиотек, таких как NCCL (NVIDIA Collective Communications Library).

Для запуска моделей llama3 или sqlcoder использовалась команда «text-generation-launcher --sharded false --num-shard=2 --max-input-length 4096 --max-total-tokens 8192 --model-id defog/sqlcoder-70b-alpha --port 8098», где --num-shard указывает количество GPU для шардинга модели, --model-id – уникальный идентификатор модели на ресурсе Huggingface<sup>20</sup> либо путь к локальной папке с файлами модели, --port – сетевой порт для обращения к TGI через протокол http.

Для генерации отправки промпт-инструкции и получения ответа БЯМ можно использовать любой http-клиент либо программную библиотеку. Например, команда «curl localhost:8080/v1/chat/completions -X POST -d ‘{“messages”: <message>, “max\_tokens”: 2048}’ -H ‘Content-Type: application/json’», где <message> – текст в json-формате (рис. 8).

В разработанной КВО все обращения к TGI реализованы программно на языке Python. Также разработанная система имеет возможность взаимодействия с открытыми API веб-порталов ChatGPT<sup>21</sup> и GigaChat<sup>22</sup>.

Для разработки серверной части КВО был использован следующий технологический стек: Python 3.10, FastAPI, Transformers, Pytorch, CUDA 12.4. БЯМ размещены на графическом сервере с 6xGPU Nvidia A100, 80 Гб VRAM.

### Клиентская часть системы

Клиент КВО выполнен в виде веб-сайта<sup>23</sup> (рис. 11) в сети Интернет, со следующей функциональностью: аутентификация и авторизация пользователя посредством Telegram, создание/удаление чатов, справочная информация о системе, ввод пользовательского запроса вручную или голосом, повторение ответов системы, вывод ответов в табличном виде, переключение между выводом в текстовом или табличном виде. Также клиент разработанной системы доступен в виде Telegram-бота<sup>24</sup>.

Для разработки был использован следующий технологический стек: ReactJS, Redux Toolkit, Styled-components.

Таким образом, подробно рассмотренное в предыдущих разделах описание КВО можно представить в следующих диаграммах взаимодействия основных компонентов (рис. 12, 13).

<sup>20</sup> <https://huggingface.co/>

<sup>21</sup> <https://chatgpt.com/>

<sup>22</sup> <https://giga.chat/>

<sup>23</sup> <http://eco-expert.ict.nsc.ru/>

<sup>24</sup> [https://t.me/eco\\_expert\\_bot](https://t.me/eco_expert_bot)

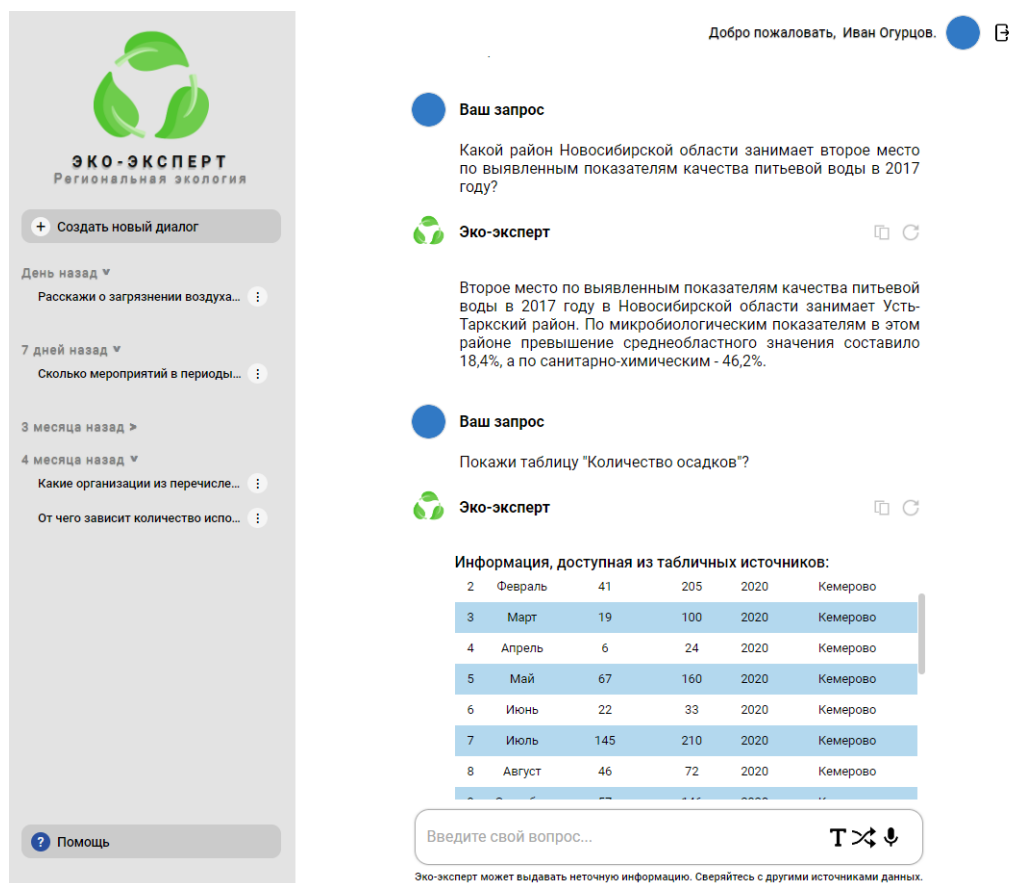


Рис. 11. Фрагмент графического интерфейса разработанной КВО  
Fig. 11. Fragment of the graphical interface of the developed KVO

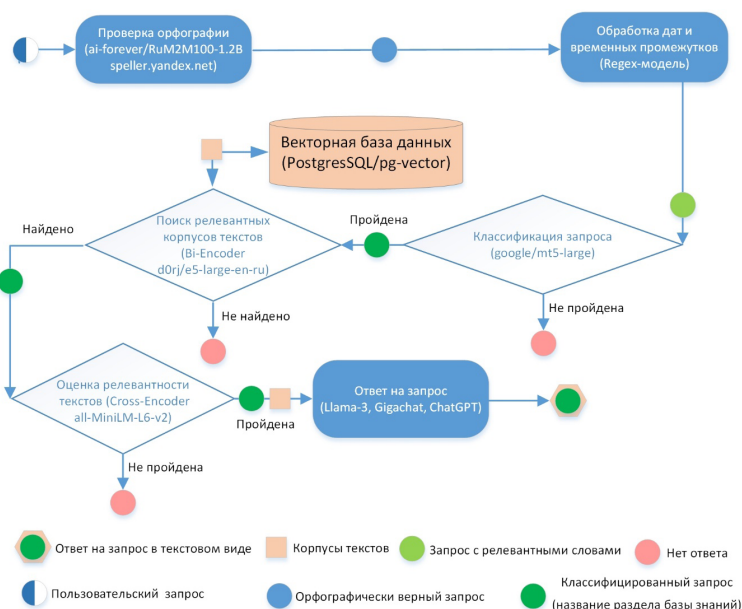


Рис. 12. Схема информационных потоков в RAG-системе для ответа LLM в текстовом виде  
передать названия БЯМ-моделей

Fig. 12. Diagram of information flows in a RAG system for LLM response in textual format with modified names of LLM models

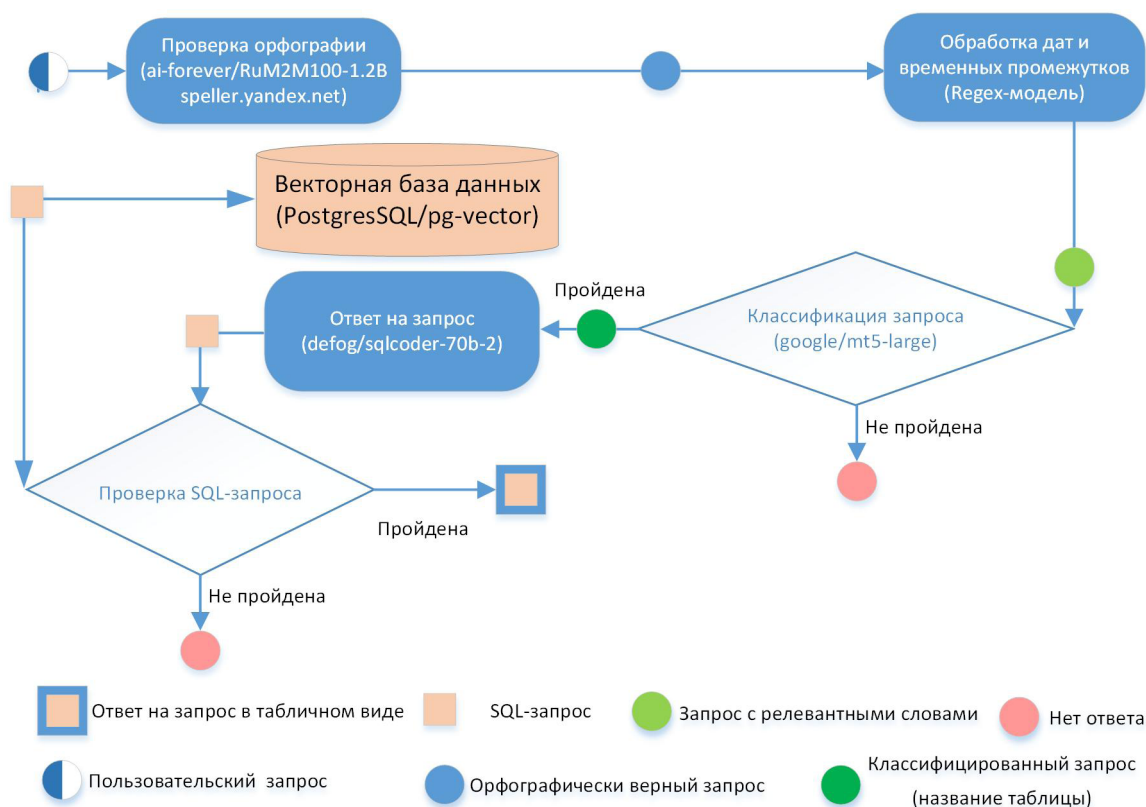


Рис. 13. Схема информационных потоков в RAG-системе для ответа LLM в табличном виде

Fig. 13. Diagram of information flows in a RAG system for LLM response in tabular format

## Метрики и тестирование системы

Для тестирования разработанной КВО-системы использовались следующие метрики:

### Bi-Encoder

Precision@k – доля релевантных документов среди первых k-предсказанных.

$$\text{Precision@k} = \frac{\text{Количество релевантных документов в top} - k}{k}$$

Recall@k – доля релевантных документов, найденных среди первых k-предсказаний.

$$\text{Precision@k} = \frac{\text{Количество релевантных документов в top} - k}{\text{Общее количество релевантных}}$$

### Cross-Encoder

MRR измеряет, насколько высоко в списке ранжированных результатов находится первый релевантный элемент. Метрика используется для задач ранжирования, где важен порядок результатов, например, в информационном поиске (1):

$$MRR = \frac{1}{N} \sum_{i=1}^N \frac{1}{rank_i}, \quad (1)$$

где  $N$  – общее количество запросов,  $rank_i$  – позиция первого релевантного документа для  $i$ -го запроса.

Метрика NDCG измеряет качество ранжирования, учитывая как порядок, так и степень релевантности документов. Нормализованная версия метрики позволяет сравнивать результаты между запросами с разным числом релевантных документов (2):

$$NDCG = \frac{DCG}{IDCG}, IDCG = \sum_{i=1}^P \frac{rel_i^{sorted}}{\log_2(i+1)}, DCG = \sum_{i=1}^P \frac{rel_i}{\log_2(i+1)}, \quad (2)$$

где  $P$  – число результатов,  $rel_i$  – релевантность документа на позиции,  $rel_i^{sorted}$  – релевантности документов, отсортированные по убыванию.

MAP (Mean Average Precision) – средняя точность всех запросов, учитывая все релевантные документы:

$$MAP = \frac{1}{N} \sum_{j=1}^N AP_j,$$

где  $AP_i$  – качество ранжирования релевантных документов для конкретного запроса.

$$AP_i = \frac{\sum_{k=1}^n Precision@k \cdot Relevant_k}{\text{Количество релевантных документов для запроса}_i},$$

где  $n$  – общее количество документов в ранжированном списке,  $Precision@k$  – точность на  $k$ -й позиции (доля релевантных документов среди первых  $k$  документов).  $Relevant@k$  – индикаторная функция, равная 1, если  $k$ -й документ релевантен, и 0 в противном случае, количество релевантных документов для запроса  $i$  – общее число релевантных документов для данного запроса.

### Лlama3 (оценка ответов)

**BERTScore** – метрика, основанная на представлениях из модели BERT, которая оценивает семантическое сходство между сгенерированным текстом и эталонным ответом.

**ROUGE-L** – метрика, которая оценивает перекрытие длинных последовательностей (longest common subsequence) между сгенерированным текстом и эталонным ответом.

**METEOR** – метрика, которая учитывает стемминг, синонимию и словоформы для более точного сравнения текстов.

Таблица 3

Сводная таблица метрик для КВО-системы

Table 3

Summary table of metrics for the CQA-system

| Метрика     | Вi-энкодер | Cross-энкодер | Llama3 |
|-------------|------------|---------------|--------|
| Precision@5 | 0,85       | 0,92          | –      |
| Recall@10   | 0,78       | 0,93          | –      |
| MRR         | 0,89       | 0,95          | –      |
| nDCG@10     | 0,87       | 0,93          | –      |
| MAP         | –          | 0,92          | –      |
| F1-score    | –          | 0,91          | –      |
| BERTScore   | –          | –             | 0,85   |
| ROUGE-L     | –          | –             | 0,72   |
| METEOR      | –          | –             | 0,43   |

Примечание. Для модели Llama3 сравнивалось 345 ответов, составленных человеком и БЯМ.

## Выводы

В рамках настоящего исследования разработана контекстно-вопросно-ответная система, основанная на современном подходе, использующем большие языковые модели и Retrieval-Augmented Generation (RAG). Реализованная система продемонстрировала значительное улучшение точности и эффективности ответов на вопросы, благодаря интеграции передовых алгоритмов обработки текста и механизмов поиска релевантной информации.

Использование Bi- и Cross-энкодеров обеспечило качественную обработку контекстуальной информации и более точную интерпретацию сложных запросов, что позволило системе адаптироваться к широкому спектру сценариев взаимодействия. Эти технологии оптимизировали процесс выбора наиболее релевантного ответа, повысив общую производительность системы.

Применение дополнительных фильтров предварительной классификации вопросов по разделам базы знаний значительно улучшило точность поиска. Этот подход минимизировал вероятность ошибок, возникающих при подборе ответа, и сократил вычислительные затраты.

Конвертация временных промежутков в явные названия месяцев и годов позволила устранить неоднозначность при обработке временных запросов. Данный подход улучшил пользовательский опыт и повысил точность ответов на вопросы, связанные с хронологией.

Использование векторной базы данных обеспечило хранение больших объемов информации и быстрый доступ к ним. Векторизация данных позволила ускорить поиск релевантных документов, что критически важно для систем, работающих с масштабируемыми наборами данных.

Применение RAG-подхода, дополненного методами классификации, векторизацией данных и визуализацией, доказало свою эффективность для решения широкого круга задач.

В перспективе система может быть расширена за счет интеграции новых алгоритмов машинного обучения, оптимизации графического интерфейса и увеличения объемов обрабатываемых данных. Эти направления могут открыть возможности для использования системы в новых прикладных областях, таких как медицина, юриспруденция, образование и пр.

## Список литературы / References

1. **Jansen B. J., Rieh S.** The Seventeen Theoretical Constructs of Information Searching and Information Retrieval. *Journal of the American Society for Information Sciences and Technology*, 2010, vol. 61(8), pp. 1517–1534. DOI: 10.1002/asi
2. **Wang H., Qin Z., Wan T.** Text generation based on generative adversarial nets with latent variables. *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018*, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22. – Springer International Publishing, 2018, pp. 92–103. <https://doi.org/10.48550/arXiv.1712.00170>
3. **Roumeliotis K. I., Tselikas N. D.** ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet*, 2023, vol. 15(6), p. 192. <https://doi.org/10.3390/fi15060192>
4. **Vidivelli S., Ramachandran M., Dharunbalaji A.** Efficiency-Driven Custom Chatbot Development: Unleashing LangChain, RAG, and Performance-Optimized LLM Fusion. *Computers, Materials and Continua*, 2024, vol. 80, iss. 2, pp. 2423–2442.

## Информация об авторах

**Попов Семен Евгеньевич**, кандидат технических наук, старший научный сотрудник

**Потапов Вадим Петрович**, доктор технических наук, профессор, главный научный сотрудник

**Замараев Роман Юрьевич**, кандидат технических наук, старший научный сотрудник

**Information about the Authors**

**Semyon E. Popov**, Candidate of Technical Sciences, Senior Researcher

**Vadim P. Potapov**, Doctor of Technical Sciences, Professor, Chief Researcher

**Roman Yu. Zamaraev**, Candidate of Technical Sciences, Senior Researcher

*Статья поступила в редакцию 10.01.2025;  
одобрена после рецензирования 27.02.2025; принята к публикации 27.02.2025*

*The article was submitted 10.01.2025;  
approved after reviewing 27.02.2025; accepted for publication 27.02.2025*

Научная статья

УДК 621.45.034.3

DOI 10.25205/1818-7900-2025-23-1-67-87

## Об особенностях моделирования распада струи жидкости на капли в потоке газа

Сергей Григорьевич Чёрный<sup>1,2</sup>  
Даниэль Тарраф<sup>2</sup>

<sup>1</sup>Институт теплофизики им. С. С. Кутателадзе СО РАН  
Новосибирск, Россия

<sup>2</sup>Новосибирский государственный университет  
Новосибирск, Россия

sgcherny2023@yandex.ru  
d.tarraf@g.nsu.ru

### Аннотация

Решается задача первичного распада струи жидкости в потоке газа и последующего ее полного распыла в двухфазном приближении. Несущая фаза – газ, дисперсная – жидкость и ее капли, образующиеся в результате распада. Для решения используется VOF-модель (Volume of Fluid), основанная на эйлерово-эйлеровом подходе, реализованная с помощью ПК ANSYS [1]. В VOF-модели перенос каждой из фаз описывается их объемными долями – непрерывными функциями от времени и пространственных переменных. В уравнении сохранения импульса смеси воздействие жидкости и газа на смесь описывается силой поверхностного натяжения, определяемой как функция кривизны и нормального вектора к границе между жидкостью и газом. VOF-модель описывает первичный распад струи жидкости. Модели смеси (Mixture) и эйлерова (Eulerian) из той же эйлерово-эйлеровой группы моделей ПК ANSYS также способны описывать первичный распад струи жидкости, но для их замыкания нужно значение среднего диаметра капель жидкой фазы (характерный размер частиц дисперсной фазы). Для описания первичного распада струи жидкости невозможно использовать эйлерово-эйлеров подход, основанный на конвективно-диффузионных уравнениях для концентрации, массы и импульса частиц (ЭЭКД) [2; 3]. Однако ЭЭКД с меньшими, чем VOF-модель, вычислительными затратами описывает распыл, стартуя от области полного распыла. Область полного распыла жидкости требуется для старта эйлерово-лагранжева подхода [4], в котором дисперсная фаза описывается путем отслеживания траекторий капель по всей расчетной области. Траектории капель вычисляются на поле течения несущей фазы, полученном из уравнений Навье – Стокса. Таким образом, полный распыл нужен для моделей с коэффициентами, зависящими от характерного размера частиц дисперсной фазы. В двухшаговых методах на первом шаге полный распыл находится по модели, описывающей первичный распад. На втором – экономичная модель стартует от полного распыла. В статье для нахождения полного распыла струи жидкости используется VOF-модель, дающая распределение объемной доли дисперсной фазы. Для ее интерпретации в терминах капель предлагается метод, верифицированный и валидированный на задачах распада жидкой пленки [5; 6] и распыла керосина в смесительном канале газовых турбин [7].

### Ключевые слова

первичный распад жидкости, полный распыл жидкости, моделирование распада жидкости

### Для цитирования

Чёрный С. Г., Тарраф Д. Об особенностях моделирования распада струи жидкости на капли в потоке газа // Вестник НГУ. Серия: Информационные технологии. 2025. Т. 23, № 1. С. 67–87. DOI 10.25205/1818-7900-2025-23-1-67-87

© Чёрный С. Г., Тарраф Д., 2025

# On the Features of Modeling the Primary Breakup of a Liquid Jet into Droplets in a Gas Flow

Sergey G. Cherny <sup>1,2</sup>, Daniel Tarraf <sup>2</sup>

<sup>1</sup>Kutateladze Institute of Thermophysics SB RAS,  
Novosibirsk, Russian Federation

<sup>2</sup>Novosibirsk State University,  
Novosibirsk, Russian Federation

sgcherny2023@yandex.ru  
d.tarraf@g.nsu.ru

## Abstract

The task of primary breakup of a liquid jet in a gas flow and its subsequent complete atomization in a two-phase approximation is being solved. The carrier phase is gas, and the dispersed phase is liquid and its droplets formed as a result of breakup. The VOF (Volume of Fluid) model, based on the Euler-Euler approach, implemented using ANSYS software [1], is used for the solution. In the VOF model, the transport of each phase is described by their volume fractions—continuous functions of time and spatial variables. In the momentum conservation equation for the mixture, the interaction of liquid and gas with the mixture is described by the surface tension force, determined as a function of the curvature and normal vector to the interface between liquid and gas. The VOF model describes the primary breakup of the liquid jet. The Mixture and Eulerian models, from the same Euler-Euler group of ANSYS models, are also capable of describing the primary breakup of the liquid jet, but they require the average droplet diameter of the liquid phase (characteristic size of the dispersed phase particles) for their closure. It is impossible to use the Euler-Euler approach, based on convective-diffusion equations for concentration, mass, and momentum of particles (EECD) [2; 3], to describe the primary breakup of the liquid jet. However, EECD, with lower computational costs than the VOF model, describes atomization, starting from the complete atomization region. The complete atomization region of the liquid is required for the start of the Euler-Lagrange approach [4], in which the dispersed phase is described by tracking the trajectories of droplets throughout the computational domain. The droplet trajectories are calculated in the flow field of the carrier phase, obtained from the Navier-Stokes equations. Thus, complete atomization is needed for models with coefficients depending on the characteristic particle size of the dispersed phase. In two-step methods, the first step finds complete atomization using a model that describes primary breakup. The second step starts from complete atomization using an economical model. In the article, the VOF model is used to find the complete atomization of the liquid jet, providing the volume fraction distribution of the dispersed phase. A method is proposed for its interpretation in terms of droplets, verified and validated on problems of liquid film breakup [5; 6] and kerosene atomization in gas turbine mixing channels [7].

## Keywords

the breakup of liquid, fully broken up liquid, modeling the breakup of liquid

## For citation

Cherny S. G., Tarraf D. On the features of modeling the primary breakup of a liquid jet into droplets in a gas flow. *Vestnik NSU. Series: Information Technologies*, 2025, vol. 23, no. 1, pp. 67–87 (in Russ.) DOI 10.25205/1818-7900-2025-23-1-67-87

## 1. Введение

### 1.1. Моделирование распыла жидкости в потоке газа

С позиций физики процесс распыла жидкости в потоке газа состоит из множества меняющихся в пространстве и во времени картин: струя жидкости; сгустки (фрагменты) жидкости; фрагменты полурегулярного размера и формы; наконец капли, близкие по геометрии к сферическим формам и, возможно, с достаточно широким спектром диаметров.

С точки зрения математического моделирования взаимодействия струи жидкости с потоком газа рассматриваются два этапа процесса взаимодействия: первичный распад струи жидкости и полный ее распыл. В этом разбиении возможно более-менее строго охарактеризовать только второй этап – полный распыл струи жидкости, который представляет собой формирование семейства четко обозначенных капель жидкости с некоторым спектром их диаметров.

В соответствии с приведенным разделением процесса распыла струи жидкости разграничим также математические модели на модели, обладающие способностью описать первичный распад и получить полный распыл, и модели, способные только стартовать от состояния полного распыла жидкости.

### *1.1.1. Эйлерово-эйлеровы модели, описывающие первичный распад жидкости*

VOF-модель (Volume of Fluid), основанная на эйлерово-эйлеровом подходе, реализованная с помощью ПК ANSYS [1], может описывать первичный распад струи жидкости и последующий полный ее распыл. Модели смеси (Mixture) и эйлерова (Eulerian) из той же эйлерово-эйлеровой группы моделей ПК ANSYS также способны описывать первичный распад струи жидкости, но для их замыкания нужно значение среднего диаметра капель жидкой фазы (характерный размер частиц дисперсной фазы).

В VOF-модели используется концепция объемной доли жидкости и смеси фаз для отслеживания жидкости во время ее распада и последующей эволюции. Смесь фаз представляет собой комбинацию жидкой и газовой фаз, а ее физические свойства, такие как плотность и вязкость, являются взвешенными средними соответствующих свойств жидкой и газовой фаз, основанными на объемных долях.

В уравнении сохранения импульса смеси воздействие жидкости и газа на смесь описывается силой поверхностного натяжения, определяемой как функция кривизны и нормального вектора к границе между жидкостью и газом. Существуют подходы [8; 9], которые связывают кривизну и нормальный вектор интерфейса с градиентом объемной доли жидкости, такие как модель континуального поверхностного натяжения (CSF), модель острого поверхностного натяжения (SSF) и сглаженная модель CSF. Наличие достаточно измельченной сетки крайне важно для успешного применения модели VOF, так как она не только улучшает точность представления жидкости, но и необходима для правильной оценки силы поверхностного натяжения в уравнении баланса импульса смеси фаз. Обеспечение высокой степени измельченности сетки может потребовать значительных вычислительных ресурсов, особенно в больших расчетных областях.

### *1.1.2. Эйлерово-эйлеровы модели, описывающие первичный распад жидкости, замыкаемые средним диаметром капель жидкой фазы (характерный размер частиц дисперсной фазы)*

Модели смеси (Mixture) и эйлерова (Eulerian) из эйлерово-эйлеровой группы моделей ПК ANSYS [1] также способны описывать первичный распад струи жидкости, но для их замыкания нужно значение среднего диаметра капель жидкой фазы (характерный размер частиц дисперсной фазы). В эйлеровой модели вместо концепции смеси фаз каждая фаза отслеживается отдельно, а также учитываются взаимодействия между ними (обмен импульсом и энергией). Это достигается решением уравнения переноса для объемной доли жидкости (нет необходимости решать его для газа в двухфазной модели) и уравнений сохранения импульса и энергии для каждой фазы. Уравнения импульса и энергии для фазы в эйлеровой модели включают термины, представляющие обмен импульсом и энергией между фазами. Эти термины содержат коэффициенты, которые являются функциями характерного размера частиц дисперсной фазы, например, в формуле Науманна [10] при оценке коэффициента обмена импульсом или в формуле Ранца – Маршалла [11; 12] при оценке коэффициента теплообмена. Этот подход эйлеровой модели не требует такого измельчения сетки, как VOF-модель, что делает его более экономичной по сравнению с VOF-моделью.

Модель смеси (Mixture) подобно VOF-модели решает уравнение переноса для объемной доли жидкости, а также уравнения сохранения импульса и энергии для фазовой смеси. Однако, в отличие от VOF-модели, модель смеси использует скорость жидкой фазы вместо скорости смеси в уравнении переноса для объемной доли жидкости. Кроме того, для представления эффектов взаимодействия жидкости и газа на фазовую смесь сила поверхностного натяжения не включается в уравнение сохранения импульса. Вместо этого рассчитывается скорость дрейфа, и соответствующий межфазный перенос импульса добавляется в уравнения сохранения. Скорость дрейфа между жидкой и газовой фазами определяется как относительная скорость жидкой фазы по отношению к газовой фазе. Скорость дрейфа может быть смоделирована и требует замыкания. Модели замыкания включают коэффициенты, которые являются функциями характерного размера частиц дисперсной фазы, как, например, в формуле Науманна [10]. Чтобы вычислить скорость жидкой фазы, используются скорости смеси и дрейфа.

### *1.1.3. Эйлерово-эйлеровы модели, не описывающие первичный распад жидкости (ЭЭКД)*

Эти модели описывают только эволюцию полного распыла жидкости и при этом стартуют от области, содержащей полностью распыленную жидкость. В [2; 3] трехмерная ЭЭКД была предложена и реализована на платформе FlowVision для моделирования процесса распыла и испарения жидких углеводородов в реакторе технического углерода. Модель включает две фазы: дисперсную фазу, представляющую распыленные жидкие капли углеводорода, и несущую фазу, состоящую из комбинации газов (теплоноситель и продукты испарения жидкости). Движение фазы-теплоносителя описывается уравнениями Навье – Стокса, а конвективно-диффузионные уравнения концентрации частиц, импульса и энергии используются для описания динамики дисперсной фазы. Требуемое распределение диаметров капель на сопле устанавливается на основе экспериментально полученных данных, которые связывают объемное распределение капель по их диаметрам.

### *1.1.4. Эйлерово-лагранжевы модели*

В эйлерово-лагранжевых моделях дисперсная фаза описывается путем отслеживания траекторий капель по всей расчетной области. Траектории капель вычисляются на поле течения несущей фазы, полученным путем решения уравнений Навье – Стокса. Таким образом, эйлерово-лагранжевы модели не описывают первичный распад жидкости. Но экономично предсказывают эволюцию полностью распыленной жидкости. Распределение размеров капель на выходе сопла может быть определено экспериментально, как это делается в [2; 3], или с использованием моделей, которые предсказывают распределение размеров капель. В [4] эйлерово-лагранжева модель использовалась для описания распыла и горения метанола в камере сгорания. Уравнения Навье – Стокса (RANS) применялись для газовой фазы, в то время как каждая капля метанола имела свой собственный набор уравнений для отслеживания ее размера, траектории и других характеристик до момента горения. Распределение размера капель на выходе сопла подчиняется закону Розина – Раммлера [13]:

$$R(x) = \exp \left( - \left( \frac{x}{d_0} \right)^q \right),$$

где  $R(x)$  – кумулятивная массовая доля частиц диаметром, меньшим или равным  $x$ ;  $d_0$  – наиболее вероятный диаметр распределения капель, а  $q$  – параметр распространения (также известный как индекс однородности).

Значение  $q$  было установлено, как у Шмидта [14]. Наиболее вероятный диаметр распределения капель  $d_0$  был рассчитан с использованием модели LISA. В этой модели предполагается, что на поверхности жидкой струи возникают возмущения, представленные в виде синусоидальных волн с определенной длиной волны  $\lambda$  и амплитудой  $\alpha$ , обе из которых определяются экспериментально. Эти возмущения увеличиваются в амплитуде по мере удаления от входа сопла, с ростом  $\omega$ , который также определяется экспериментально. Длина распада  $L$  оценивается как  $L = C \frac{U}{\omega}$ , где  $C$  – пропорциональная константа, определяется экспериментально, а  $U$  – осевая скорость жидкости. Когда возмущения достигают этой длины  $L$ , жидкая струя распадается на фрагменты. Диаметр фрагментов  $d_f$  в точке распада определяется с использованием баланса масс, обеспечивающим равенство масс жидкости до и после распада  $d_f = \sqrt{\frac{8\rho U^2}{\sigma k}}$ , где  $\rho$  – плотность жидкости,  $\sigma$  – поверхностное натяжение,  $k = \frac{2\pi}{\lambda}$  – волновое число. Распад фрагментов на капли происходит согласно анализу Вебера для капиллярной неустойчивости. Следовательно, наиболее вероятный диаметр распределения капель определяется из  $d_0 = 1,88 d_f (1 + 3Oh)^{\frac{1}{6}}$ , где  $Oh$  – число Огенсаберга.

## 1.2. Методы интерпретации распределения объемной доли жидкости в терминах ее капель

Во всех ранее упомянутых моделях, за исключением VOF-модели, требуется знание размера капель. Это может быть либо среднее значение размера капель, либо размеры и местоположения капель всего рассматриваемого семейства. Таким образом, потребность в нахождении полного распыла струи жидкости в потоке газа возникает в случаях использования моделей, в которых для их замыкания требуется характерный размер частиц дисперсной фазы. Или в двухшаговых методах, на первом шаге которых полный распыл находится по модели, описывающей первичный распад. На втором – экономичная модель стартует от полного распыла. В настоящей статье для нахождения полного распыла струи жидкости используется VOF-модель, дающая распределение объемной доли дисперсной фазы. Для ее интерпретации в терминах капель с целью дальнейшего их использования по назначению предлагается метод, верифицированный и валидированный на задачах распада жидкой пленки [5; 6] и распыла керосина в смесительном канале газовых турбин [7].

Прежде чем излагать предложенный в настоящей работе подход преобразования распределения объемной доли жидкости в семейство капель, рассмотрим три существующие разработки в этой области других авторов. Их немного.

### 1.2.1. Метод выделения и идентификации сгустков фрагментов жидкости

В [15] предлагается двухшаговый подход к описанию распада струи жидкости. На первом шаге применяется VOF-модель, которая обладает способностью описывать первичный распад и далее осуществляет переход к полному распылу жидкости. На втором шаге включается экономичная лагранжева DPM-модель, стартующая от полученного на первом шаге полного распыла. Преобразование распределения объемной доли жидкости в капли осуществляется путем выделения и идентификации сгустков фрагментов жидкости. Сначала вычисляется объемно-эквивалентный диаметр для кластеров жидкости в вычислительной области для проверки их попадания в заданный диапазон размеров. На этом этапе отсеиваются капли слишком малые и слишком великие из соображений реалистичности процесса распада струи жидкости. Этот

диапазон указывается вручную, что делает этот подход зависящим от субъективного представления о размерах капель, конкретно минимальных и максимальных размерах капель, но не всего распределения. Затем анализируется форма этих кластеров, чтобы подтвердить, что они почти сферические. Это включает два специфических критерия: стандартное отклонение нормализованного радиуса и среднюю ортогональность радиуса поверхности. Стандартное отклонение нормализованного радиуса оценивает равномерность радиуса капли, чтобы обеспечить минимальное отклонение от сферической формы. Низкое стандартное отклонение указывает на то, что радиус консистентен по всей капле, что характерно для сферической формы. Средняя ортогональность радиуса поверхности измеряет углы между точками поверхности, чтобы подтвердить почти сферическую форму капли. Высокая ортогональность указывает на то, что точки поверхности хорошо распределены вокруг центра, что также подтверждает сферическую природу.

Для проверки метода был использован двухшаговый подход (VOF + DPM) вместе с методом выделения и идентификации сгустков фрагментов жидкости для моделирования распада жидкой струи в поперечном воздушном потоке. Сравнение среднего диаметра Заутера (SMD) с экспериментальными данными было проведено. Результаты сравнения демонстрируют способность выделять и идентифицировать скопления фрагментов жидкости.

### 1.2.2. Метод пороговых значений

В работе [16] представлен метод пороговых значений для идентификации капель и их размеров в распределении объемной доле жидкости. Процесс начинается с обхода всей области мониторинга, предполагая, что в ней произошел полный распыл. При обходе идентифицируются ячейки сетки со значительной объемной долей жидкости. Для этого используются конкретные пороговые значения объемной доли жидкости, такие как  $n = 0,9; 0,5; 0,1; 0,05$ . Ячейки сетки с объемной долей, превышающей эти пороговые значения, считаются значительными. При мониторинге текущего порогового значения ячейки, идентифицированные для предыдущего порогового значения, не рассматриваются. После идентификации значительной ячейки сетки все смежные ячейки сетки с объемной долей выше включаются для формирования одного кластера из ячеек сетки, представляющего каплю. Объем каждой идентифицированной капли вычисляется с использованием формулы  $V_d = \sum_{i=1}^N V_i \alpha_i$ , где  $V_i$  – объем  $i$ -й ячейки сетки, а  $\alpha_i$  – объемная доля жидкости в  $i$ -й ячейке сетки. Эта процедура повторяется до тех пор, пока вся область мониторинга не будет охвачена, что обеспечивает идентификацию всех капель и расчет их объемов.

Авторы провели численное моделирование распада жидкой пленки, используя модель VOF. Они определили размеры капель, используя метод пороговых значений в области мониторинга, и рассчитали глобальный SMD в этой области как функцию времени. Сравнение рассчитанного глобального SMD с соответствующими экспериментальными данными во времени показывает хорошее совпадение, что указывает на способность метода выявлять капли и оценивать их размер при применении в области, в которой произошел полный распыл.

### 1.2.3. VOF droplet conversion

В работе [17] методы Volume of Fluid (VOF) и Large Eddy Simulation (LES) используются для изучения распада жидкой струи в поперечном воздушном потоке. Этот подход использует модель VOF для моделирования жидкой фазы и LES для моделирования турбулентной газовой фазы, а также метод Lagrangian Particle Tracking (LPT) для отслеживания движения и свойств капель, образующихся в процессе распада.

Переход от эйлеровой VOF к лагранжевой LPT осуществляется с использованием метода Шпитценбергера [18]. В этом методе маркируются все ячейки с объемной долей жидкости  $\alpha > \alpha_t$  как жидкостные ячейки, где  $\alpha_t$  – это пороговая объемная доля. Эти ячейки затем проходят итерацию и маркируются уникальными идентификаторами. Смежные ячейки затем маркируются наименьшим идентификатором одной жидкой капли, что приводит к идентификации капель из объемной доли жидкости VOF. После идентификации рассчитываются свойства капель, такие как объем, положение, скорость и диаметр. Для обеспечения возможности преобразования жидкой капли в лагранжевы частицы используются два критерия: размер капли должен быть меньше максимального диаметра капли 60 микрон, а сферичность  $\psi_p$  должна быть меньше 2,5. Уравнения для расчета диаметра капли и сферичности следующие:

$$d_p = \sqrt[3]{\frac{6V_p}{\pi}}, \quad \psi_p = \frac{2r_{\max}}{d_p},$$

где  $d_p$  – диаметр капли,  $V_p$  – объем капли,  $\psi_p$  – сферичность капли, а  $r_{\max}$  – максимальное расстояние от центра массы до свободной поверхности капли.

Сравнение результатов VOF/LES/метода Шпитценбергера/LPT с VOF+LES и экспериментальными данными показало хорошее согласие как для качественных характеристик (динамика распыления, формирование распыления, преобразование капель и эволюция распыления), так и для количественных свойств (проникновение и изгиб струи, осевая скорость капель и распределение размеров капель), что указывает на способность метода Шпитценбергера.

### 1.3. Цель данной работы

Цель данной работы – создание усовершенствованного метода для идентификации отдельных капель в распределении объемной доли полного распыла жидкости, полученного по VOF-модели. Назовем созданный в настоящей работе подход преобразования объемной доли жидкости в множество капель «метод локальных максимумов». Этот метод определяет центр каждой капли, находя локальные максимумы в распределении объемной доли жидкости. Эти локальные максимумы затем группируются с окружающими ячейками, имеющими монотонно убывающие значения объемной доли, для формирования кластеров ячеек, где каждый кластер представляет каплю.

Преимущества этого метода заключаются в том, что он не требует указания диапазона размеров капель, как в методе, описанном в разделе 1.2.1, и не требует пороговых значений, как в методах, описанных в разделах 1.2.2 и 1.2.3. Недостатком является то, что этот метод требует ручного определения области мониторинга, в которой предполагается полный распыл жидкости, аналогично методу, описанному в разделе 1.2.2.

### 1.4. Структура статьи

В разделе 2 представлена двухфазная модель VOF для моделирования распыла жидкости в газ. Также показана ее способность предсказывать форму распыла путем сравнения с экспериментальными данными [19]. В разделе 3 описан метод формирования капель из распределения объемной доли жидкости (метод локальных максимумов). В разделе 4 метод локальных максимумов был валидирован на двух задачах: 1) распад жидкой пленки, и 2) распыл керосина в предварительном смесительном канале для камер сгорания газовых турбин при высоком давлении. В этих задачах глобальные и локальные значения SMD были рассчитаны на основе результатов метода локальных максимумов и сравнивались с экспериментами.

## 2. VOF-модель

### 2.1. Уравнения

Для задачи распыла жидкостей используют двухфазную модель VOF, где первая фаза – жидкость, распыляемый материал, а вторая фаза – газ, среда, в которой распыляется жидкость. В этой модели существует уравнение объемной доли жидкости:

$$\frac{\partial \alpha_l}{\partial t} + \nabla \cdot (\alpha_l \mathbf{u}) = 0,$$

объемная доля газа рассчитывается алгебраически по формуле:

$$\alpha_g = 1 - \alpha_l.$$

Здесь  $\alpha_l$ ,  $\alpha_g$  – объемная доля жидкости и газа,  $\mathbf{u}$  – скорость смеси жидкости и газа. Уравнение неразрывности смеси:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0,$$

уравнение сохранения импульса смеси:

$$\frac{\partial}{\partial t} (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot \left[ \mu \left( \nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right] + \rho \mathbf{g} + \mathbf{F}_\sigma.$$

Здесь  $\rho$  – плотность смеси, рассчитанная с использованием плотностей жидкости  $\rho_l$  и плотностей газа  $\rho_g$  следующим образом:

$$\rho = \alpha_l \rho_l + \alpha_g \rho_g,$$

где  $p$  – давление,  $\mu$  – вязкость смеси, рассчитанная с использованием вязкостей жидкости  $\mu_l$  и вязкостей газа  $\mu_g$  следующим образом:

$$\mu = \alpha_l \mu_l + \alpha_g \mu_g,$$

где  $\mathbf{g}$  – гравитационное ускорение,  $\mathbf{F}_\sigma$  – сила поверхностного натяжения, моделируется следующим образом:

$$\mathbf{F}_\sigma = \sigma \frac{2\rho \nabla \hat{n}}{\rho_l + \rho_g} \nabla \alpha; \hat{n} = \frac{\nabla \alpha}{|\nabla \alpha|}.$$

### 2.2. Валидация VOF-модели

Для демонстрации способности модели VOF предсказывать форму распыла был проведен тест, аналогичный тесту в [16]. Поток воздуха входит в туннель (стены туннеля – 3, 4, 5, 6) через (1) и выходит из (2), как показано на рис. 1. Вода распыляется вертикально через сопло в поток воздуха. Под воздействием воздуха струя воды наклоняется вперед.

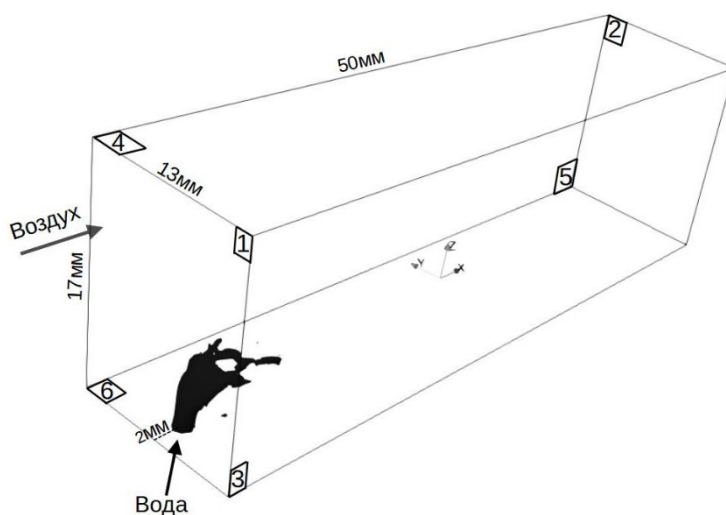


Рис. 1. Расчетная область: 1 – вход воздуха; 2 – выход; 3, 4, 5, 6 – стены туннеля

Fig. 1. Design area, 1 – air inlet; 2 – exit; 3, 4, 5, 6 – tunnel walls

Эксперименты [19] показали, что при скорости воздуха 120,4 м/с и скорости воды 10,83 м/с центральная линия водяной струи наклоняется, как показано на рис. 2 (черная сплошная линия).

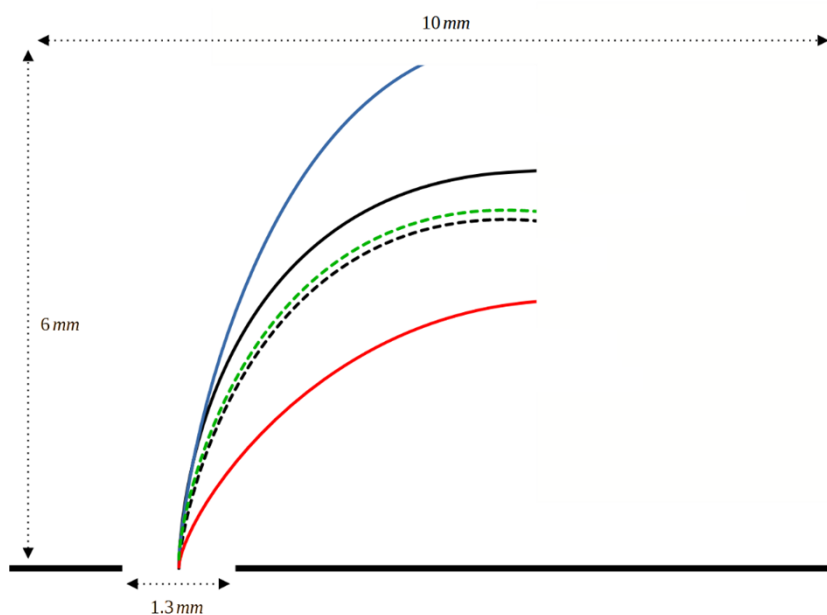


Рис. 2. Сравнение центральных линий струй: эксперименты – черная сплошная линия; VOF – черная пунктирная линия; (Eulerian,  $d = 10^{-4}$ ) – сплошная синяя линия; (Eulerian,  $d = 3 \times 10^{-4}$ ) – пунктирная зеленая линия; (Eulerian,  $d = 10^{-5}$ ) – сплошная красная линия

Fig. 2. Comparison of the central lines of the jets: Experiments – black solid line; VOF – black dotted line; (Eulerian,  $d = 10^{-4}$ ) – solid blue line; (Eulerian,  $d = 3 \times 10^{-4}$ ) – dotted green line; (Eulerian,  $d = 10^{-5}$ ) – solid red line

На рис. 2 также показаны центральные линии жидкой струи, полученные в результате моделирования с использованием модели VOF (черная линия) и эйлеровой модели из [20]

для трех различных значений среднего размера капель:  $10^{-5}$  м – сплошная красная линия,  $3 \times 10^{-4}$  м – пунктирная зеленая линия,  $10^{-4}$  м – сплошная синяя линия.

### 3. Метод локальных максимумов

Целью данного метода является идентификации отдельных капель в объемной доле полностью распавшейся жидкости  $\alpha_l(x, y, z, t)$  модели VOF, как показано на рис. 3.

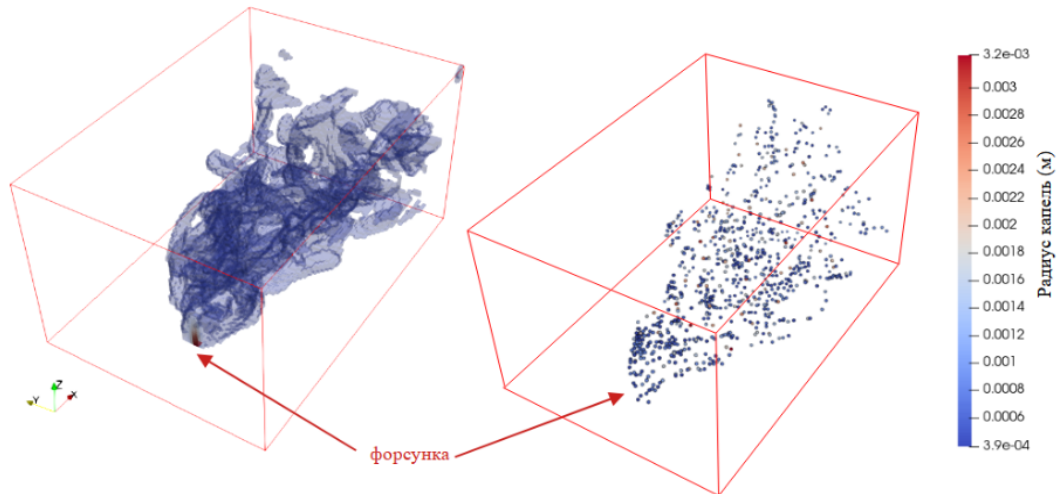


Рис. 3. Вывод местоположения и размеров капель из распределения объемной доли жидкости:  
слева – распределение объемной доли жидкости  $\alpha_l$ , полученное с помощью модели VOF;  
справа – расположение и размеры капель (результаты метода)

Fig. 3. Deduction of the location and size of droplets from the distribution of the volume fraction of the liquid.

On the left is the distribution of the volume fraction of the liquid  $\alpha_l$ , obtained using the VOF model;

On the right is the location and size of the droplets (the results of the method)

Метод основан на предположении, что, во-первых, жидкость полностью распалась; и, во-вторых, центр каждой капли является локальным максимумом в распределении объемной доли жидкости, и ячейка, которая содержит этот локальный максимум, вместе со всеми окружающими ячейками, которые содержат монотонно убывающие значения объемной доли жидкости  $\alpha_l$ , образуют группу ячеек, которая представляет каплю, соответствующую данному локальному максимуму. Для каждого локального максимума будет найдена соответствующая группа ячеек, на основе которой будет рассчитан центр и размер соответствующей капли. Предположим, что существует  $N$  групп ячеек (или капель), и в каждой группе существует  $n_i$  ячеек, где  $i = 1, \dots, N$ . Центр капли можно рассчитать следующим образом:

$$x_i = \frac{1}{A_i} \sum_{j=1}^{n_i} (\alpha_l)_j (x_c)_j,$$

$$y_i = \frac{1}{A_i} \sum_{j=1}^{n_i} (\alpha_l)_j (y_c)_j,$$

$$z_i = \frac{1}{A_i} \sum_{j=1}^{n_i} (\alpha_l)_j (z_c)_j,$$

$$A_i = \sum_{j=1}^{n_i} (\alpha_l)_j,$$

где  $(\alpha_l)_j$  – объемная доля жидкости ячейки  $j$  в группе  $i$ ,  $(x_c)_j$ ,  $(y_c)_j$ ,  $(z_c)_j$  – центр ячейки  $j$  в группе  $i$ . Объем капли  $i$  рассчитывается следующим образом:

$$V_i = \sum_{j=1}^{n_i} (\alpha_l)_j (V_c)_j,$$

где  $(v_c)_j$  – объем ячейки  $j$  в группе  $i$ . Предположив, что все капли имеют сферическую форму, можно рассчитать радиус капли следующим образом:

$$R_i = \sqrt[3]{\frac{3}{4\pi} V_i}.$$

При сравнении метода локальных максимумов с методом, описанным в разделе 1.2.1, последний превосходит в определении недавно сформированных капель в областях, где присутствуют и капли, и жидкостная струя. Метод 1.2.1 определяет каждую каплю на основе ее формы и предварительных знаний о максимальных и минимальных размерах капель, в то время как метод локальных максимумов не может различать струю и распавшуюся часть жидкости. В областях полного распада жидкости метод локальных максимумов работает с высокой точностью без необходимости задания параметров, в отличие от метода 1.2.1. При сравнении с методом 1.2.2 оба метода требуют полного распада струи жидкости, но метод 1.2.2 требует пороговых значений, тогда как метод локальных максимумов не нуждается в таких параметрах. Сравнивая метод локальных максимумов с методом 1.2.3, последний использует критерии размера и формы капель для определения недавно сформированных капель в областях, где имеются и капли, и струя. Метод 1.2.3 требует дополнительных коэффициентов, таких как «максимальный диаметр капли» и «сферичность капли», которые необходимо оценить, в то время как метод локальных максимумов не требует этих коэффициентов, что упрощает его реализацию.

#### 4. Валидация метода

Для проверки представленного метода будут выполнены три шага. Во-первых, будут смоделированы две проблемы распыла с использованием модели VOF: распада жидкой пленки и распыла керосина в предварительном смесительном канале. Во-вторых, на основе распределения объемной доли жидкости, полученного в результате моделирования, с помощью метода будет выведено распределение капель и их размеры. Наконец, на основе результатов метода локальных максимумов будет рассчитан средний диаметр Заутера (SMD) и сравнен с его значением, измеренным экспериментально.

Средний диаметр Заутера – это общепринятая величина, используемая для характеристики распыла:

$$SMD = \frac{6V}{S},$$

где  $V$  – общий объем всех капель,  $S$  – общая поверхность всех капель.

#### 4.1. Задача распада жидкой пленки

В этой задаче (рис. 4) вода впрыскивается из двух противоположных сопел, две струи жидкости сталкиваются и образуют водяную пленку, которая впоследствии распадается на капли.

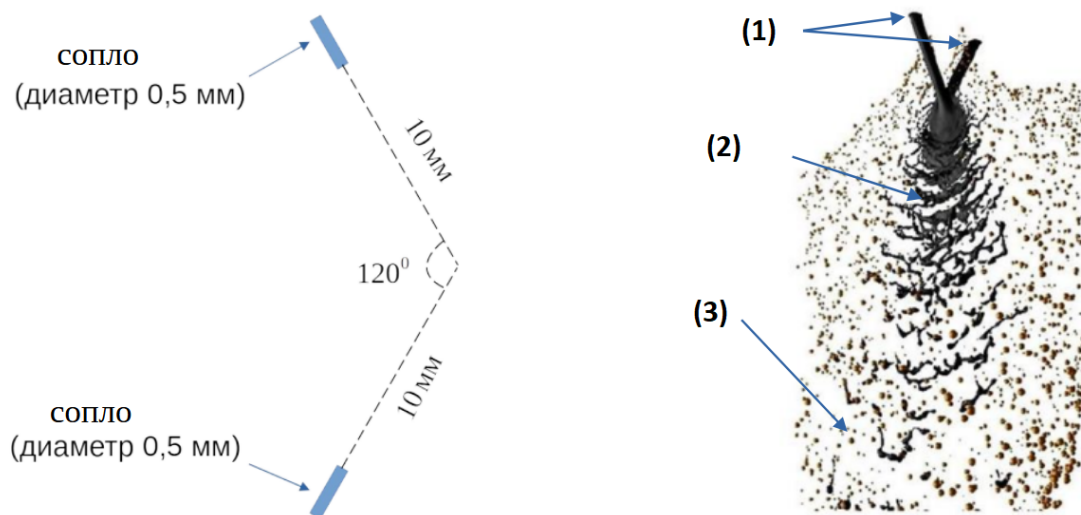


Рис. 4. Постановка задачи распада жидкой пленки: слева – настройка сопел; справа – этапы распыления:

1 – впрыск воды; 2 – образование водяной пленки; 3 – распад водной пленки на капли

Fig. 4. Formulation of the problem of disintegration of a liquid film: on the left – adjustment of nozzles;

on the right – stages of spraying: 1 – injection of water; 2 – formation of a water film;

3 – decomposition of a water film into droplets

Для моделирования задачи с использованием модели VOF, расчетная область определяется, как показано на рис. 5.

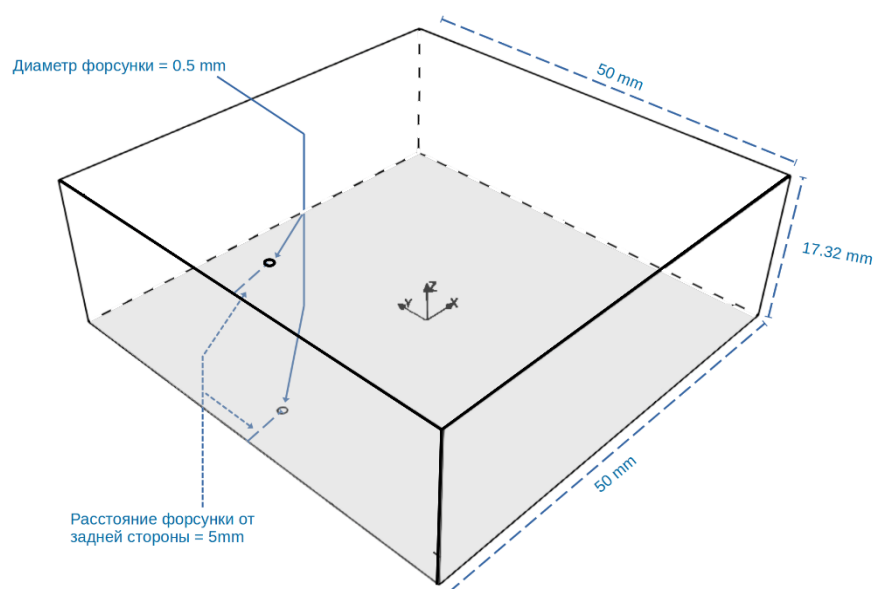


Рис. 5. Расчетная область

Fig. 5. Calculation area

В числовой задаче начальные значения внутри расчетной области устанавливаются  $\alpha = 0$ ,  $p = 101\,323$  Па и  $\mathbf{u} = 0$ . Граничные условия на боковых границах (кроме сопел) задаются как  $\frac{\partial \alpha}{\partial n} = 0$ ,  $p = 101\,325$  Па и  $\frac{\partial \mathbf{u}}{\partial n} = 0$ . На верхнем сопле  $\alpha_l = 1$ ,  $\frac{\partial p}{\partial n} = 0$ , компоненты скорости смеси —  $u_x = 2,546$  м/с,  $u_y = 0$ ,  $u_z = -4,410$  м/с. На нижнем сопле  $\alpha_l = 1$ ,  $\frac{\partial p}{\partial n} = 0$ , компоненты скорости смеси —  $u_x = 2,546$  м/с,  $u_y = 0$ ,  $u_z = -4,410$  м/с. Физические параметры устанавливаются по умолчанию в платформе ANSYS. Построенную расчетную сетку для расчетной области можно увидеть на рис. 6. Эта сетка неструктурирована и содержит начальное уточнение вблизи сопел. Причина первоначального уточнения вблизи сопел заключается в том, чтобы облегчить методу адаптации обнаружение изменения градиента объемной доли жидкости, чтобы начать адаптацию от сопел. Метод адаптации работает следующим образом: после каждого временного шага проводится проверка каждой ячейки сетки, если  $0 \leq \frac{\nabla \alpha_l}{\max |\nabla \alpha_l|} \leq S_1$ , ячейка огрубится, если  $S_1 \leq \frac{\nabla \alpha_l}{\max |\nabla \alpha_l|} \leq S_2$ , ячейка не изменится, если  $S_2 \leq \frac{\nabla \alpha_l}{\max |\nabla \alpha_l|} \leq 1$ , ячейка будет уточняться, где  $S_1 = 0,1$  и  $S_2 = 0,11$  — контролируемые уровни.

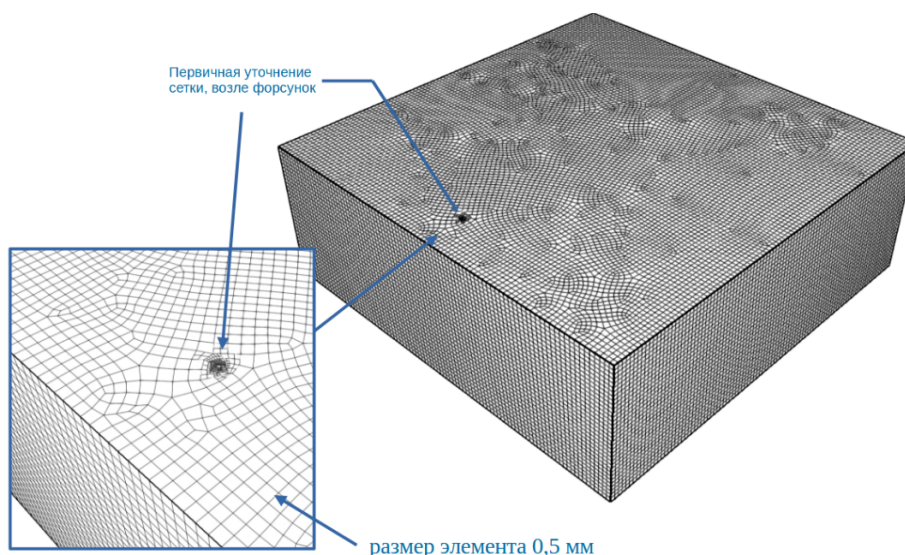


Рис. 6. Первоначальная расчетная сетка  
Fig. 6. Initial calculation grid

Распределение объемной доли воды в трех временных точках, полученное в результате моделирования с использованием модели VOF и адаптивных сеток, можно наблюдать на рис. 7. В первой точке времени ( $t = 2,6 \times 10^{-3}$  с) можно наблюдать столкновение двух струй воды и образование водяной пленки. Во второй точке времени ( $t = 4 \times 10^{-3}$  с) можно наблюдать распад водяной пленки на капли. В третьей точке времени ( $t = 6,7 \times 10^{-3}$  с) показано развитие распыла в дальнейшем времени. Продолжение расчета покажет только выход капель из границ, в то время как новые капли образуются вблизи водяной пленки.

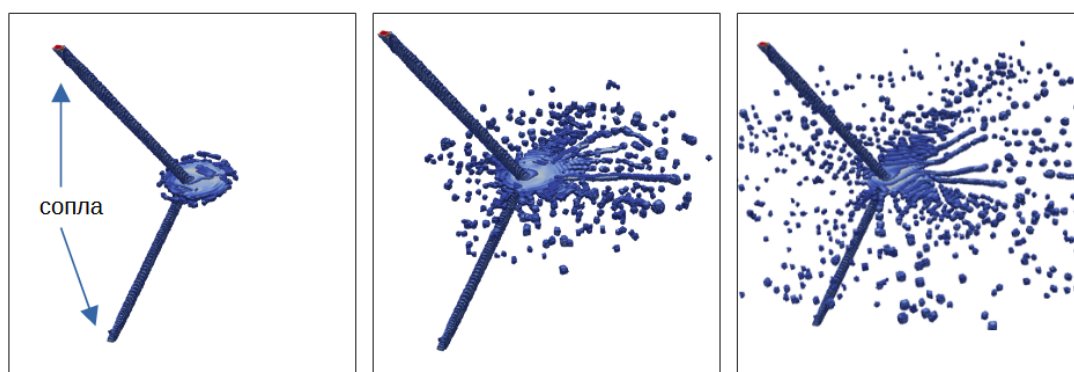


Рис. 7. Распределение объемной доли, полученное в результате моделирования с использованием модели VOF: слева – распределение в момент времени  $t = 2,6 \times 10^{-3}$  с, показывающее образование жидкой пленки после столкновения двух струй жидкости; в центре – распределение в момент времени  $t = 4 \times 10^{-3}$  с, показывающее распад жидкой пленки на капли; справа – распределение в более ранний момент времени  $t = 6,7 \times 10^{-3}$  с

Fig. 7. The volume fraction distribution obtained as a result of modeling using the VOF model; on the left is the distribution at time  $t = 2,6 \times 10^{-3}$  s, showing the formation of a liquid film after the collision of two liquid jets; in the center is the distribution at time  $t = 4 \times 10^{-3}$  s, showing the decomposition of the liquid film into droplets; on the right is the distribution at an earlier time  $t = 6,7 \times 10^{-3}$  s

Чтобы убедиться, что метод локальных максимумов работает корректно, его следует применять только к полностью распавшейся части жидкости. Распределение объемной доли жидкости на рис. 7 в его текущей форме не подходит для ввода в метод, поскольку оно содержит не только полностью распавшуюся жидкость, но также две струи и жидкую пленку (не полностью распавшиеся). Самый простой способ решить эту проблему – вручную удалить эти части из распределения, как показано на рис. 8. Все значения объемной доли жидкости внутри цилиндра установлены равными нулю. Недостатком этого подхода является то, что он удаляет некоторые капли, которые мы не собирались удалять

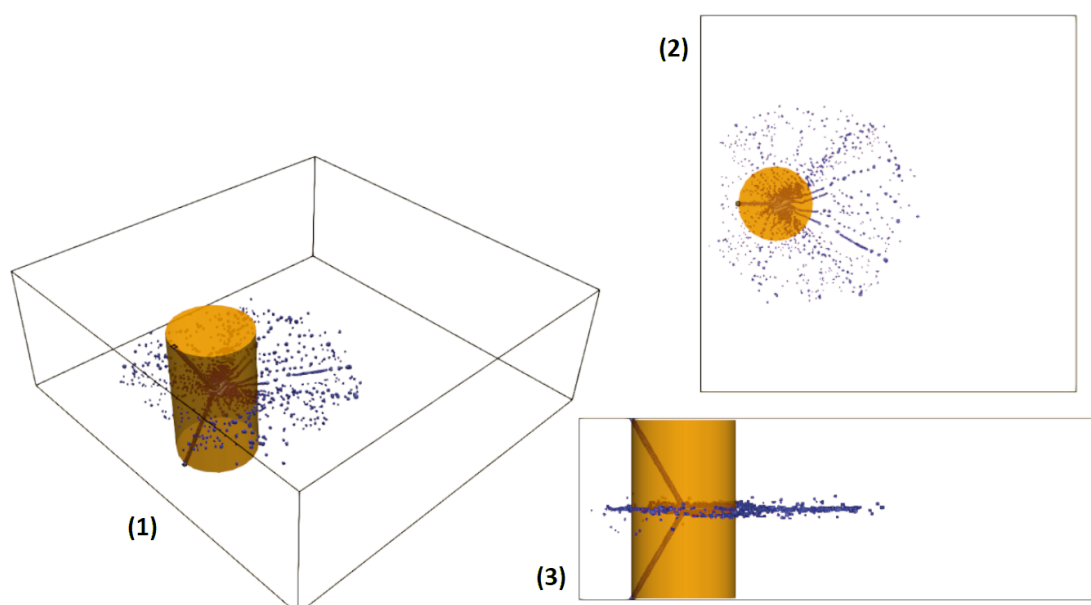


Рис. 8. Удаление не полностью распавшейся части жидкости: (1) изометрический вид, (2) вид сверху, (3) вид сбоку

Fig. 8. Removal of the incompletely decomposed part of the liquid: (1) isometric view, (2) top view, (3) side view

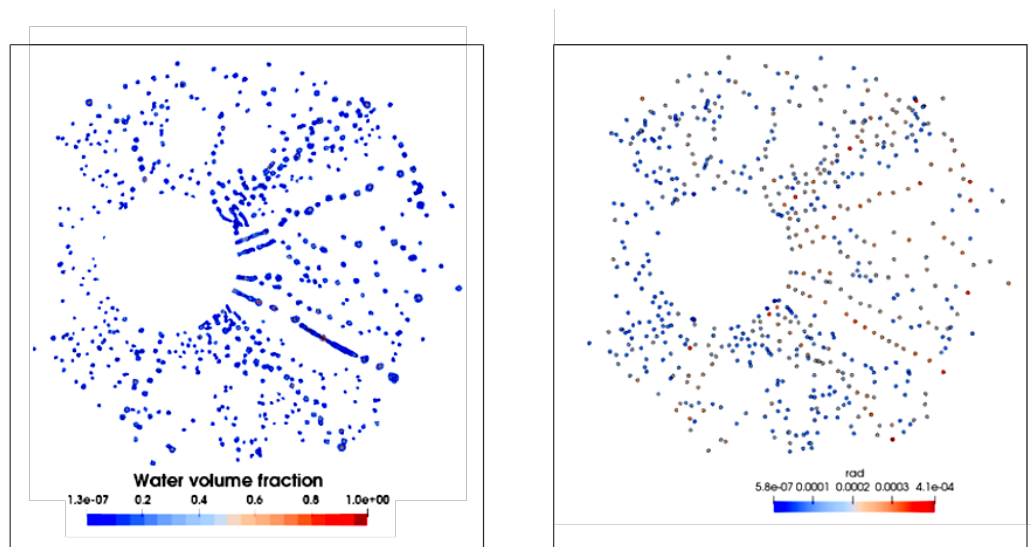


Рис. 9. Результаты применения метода локальных максимумов к объемной доле жидкости: слева – распределение объемной доли; справа – расположение и размеры капель (результаты метода)  
 Fig. 9. The results of applying the method of local maxima to the volume fraction of a liquid; on the left – the distribution of the volume fraction; on the right – the location and size of droplets (the results of the method)

Оставшаяся часть объемной доли жидкости после обрезания показана на рис. 9 слева. Результаты применения метода можно увидеть справа.

Используя данные о каплях, можно рассчитать средний диаметр Заутера (SMD), характерный для распыла. Применяя метод локальных максимумов в последовательности временных точек, можно видеть на рис. 10 сходимость глобальную значения SMD к близкому значению, полученному в результате экспериментов [5; 6], а также результаты, полученные в работе [16].

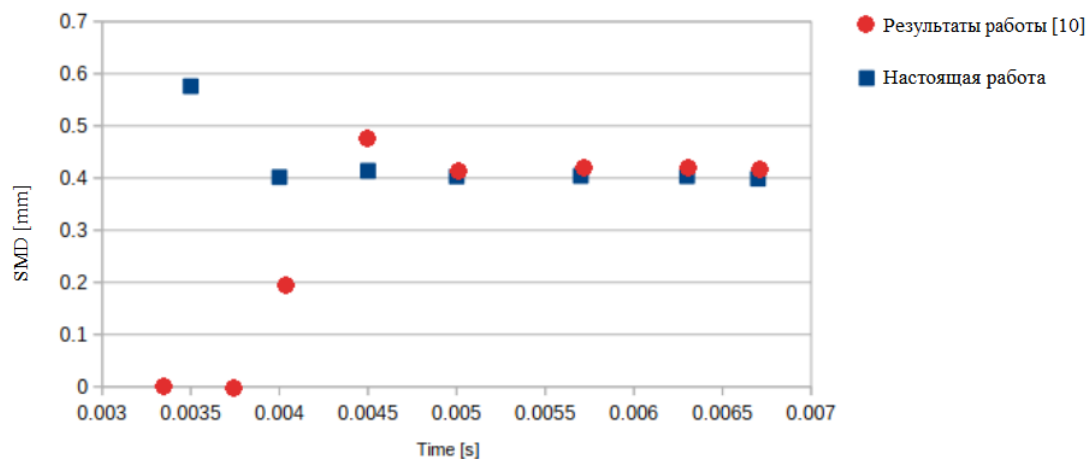


Рис. 10. Сравнение значений SMD, полученных с помощью разработанного метода локальных максимумов, и результатов, полученных из эксперимента  
 Fig. 10. Comparison of the SMD values obtained using the developed method of local maxima and the results obtained from the experiment

На рис. 10 видно, что в начале времени значение расчетного SMD заметно отличается от измеренного значения, но с течением времени они совпадают. Это происходит потому,

что эта задача решается как нестационарная задача VOF, где решение сходится в более позднее время, то же время, когда расчетные и измеренные значения SMD лучше совпадают, и это совпадение продолжается со временем. Это демонстрирует, что метод эффективно определил размеры капель, на основе которых рассчитывается SMD (в случае сходимости численного решения, по которому определяется объемная доля жидкости).

#### 4.2. Задача распыла керосина в предварительном смесительном канале для камер сгорания газовых турбин при высоком давлении

В этой задаче (рис. 11) аэрозоль керосина распыляется из одной форсунки в канал, прежде чем он испарится. Локальный SMD измерялся в 4 точках на расстоянии 30, 60, 100, 150 мм от сопла [7]. Для сравнения с этими экспериментальными измерениями тот же самый предыдущий подход будет использован для получения распределения размера капель, на основании которого будет рассчитан и сравнен локальный SMD.

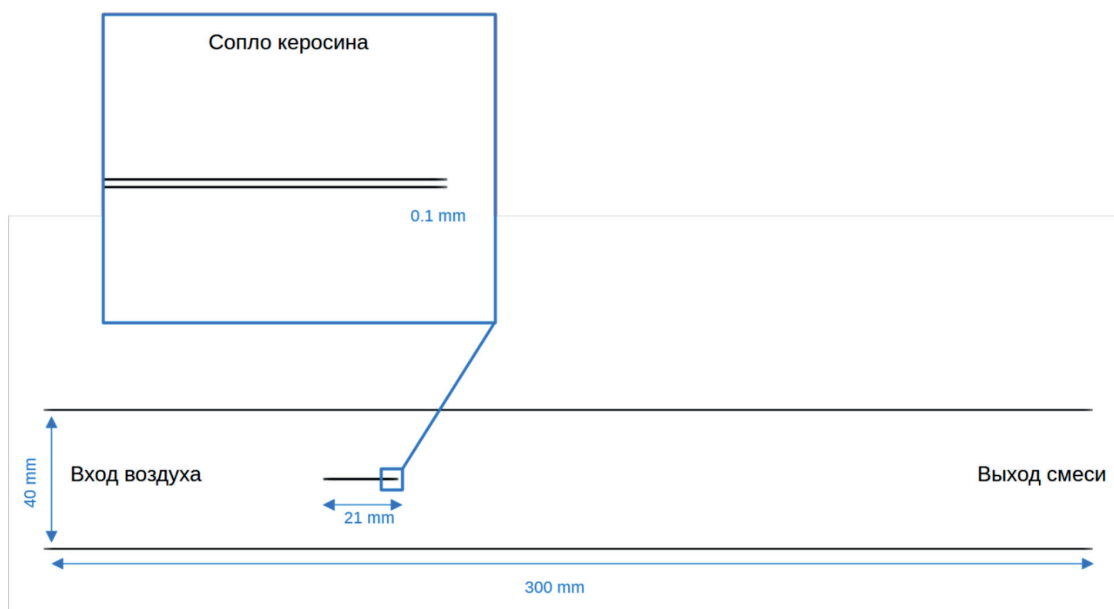


Рис. 11. Расчетная область (2D); распыл керосина в канал  
Fig. 11. Calculated area (2D); kerosene spray into the channel

В числовой задаче начальные значения внутри расчетной области: объемная доля керосина –  $\alpha_l = 0$ , давление –  $p = 101325$  Па и компоненты скорости смеси –  $u_x = 120$  м/с,  $u_y = 0$ . Граничные условия: на входе воздуха объемная доля керосина –  $\alpha_l = 0$ , давление –  $\frac{\partial p}{\partial n} = 0$ , компоненты скорости смеси –  $u_x = 120$  м/с,  $u_y = 0$ ; на сопле массовый расход керосина 1 г/с, давление –  $\frac{\partial p}{\partial n} = 0$ ; на выходе смеси объемная доля керосина –  $\frac{\partial \alpha_l}{\partial n} = 0$ , давление –  $\frac{\partial p}{\partial n} = 0$ , скорость смеси –  $\frac{\partial u}{\partial n} = 0$ . Физические параметры устанавливаются по умолчанию в платформе ANSYS.

Полученную объемную долю керосина можно увидеть на рис. 12, распределение размеров капель можно увидеть на рис. 13.

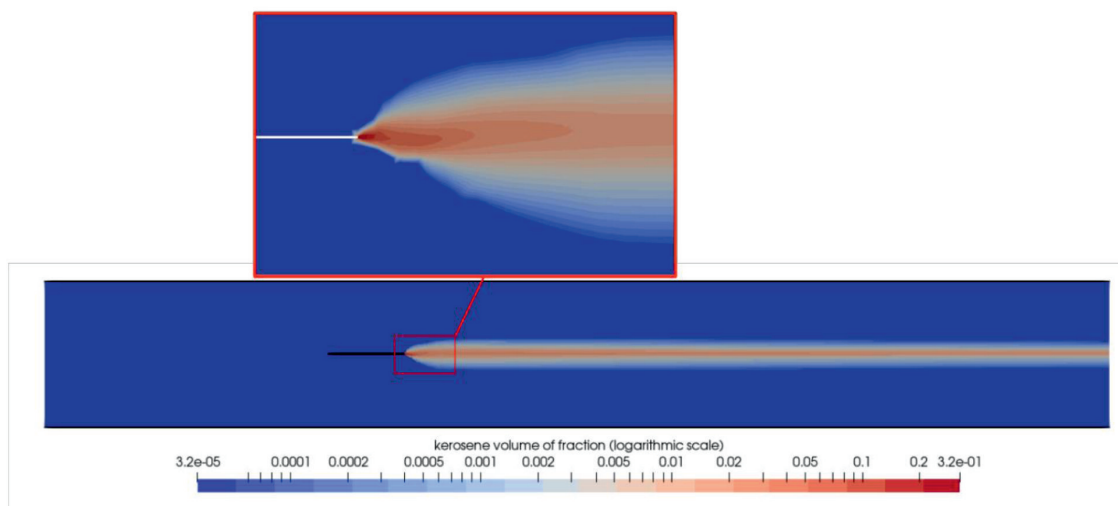


Рис. 12. Распределение объемной доли керосина

Fig. 12. Distribution of kerosene volume fraction

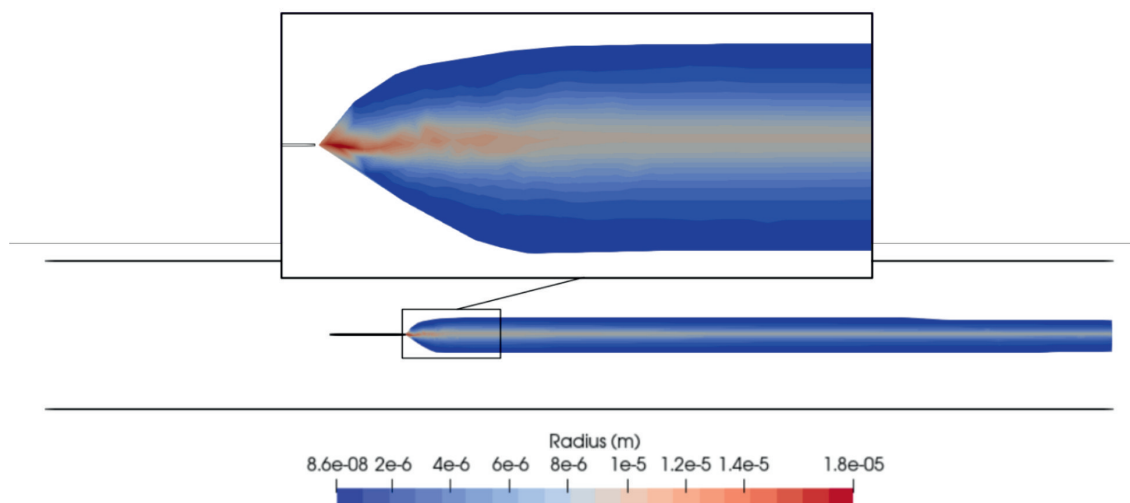


Рис. 13. Распределение размеров капель керосина, полученное с помощью метода локальных максимумов

Fig. 13. Kerosene droplet size distribution obtained using the local maxima method

Для сравнения значение SMD с экспериментальными результатами следует рассчитать SMD локально в 4 определенных точках (30, 60, 100, 150 мм от сопла). Это делается путем расчета SMD в полосах шириной 10 мм, окружающих точки, как показано на рис. 14.

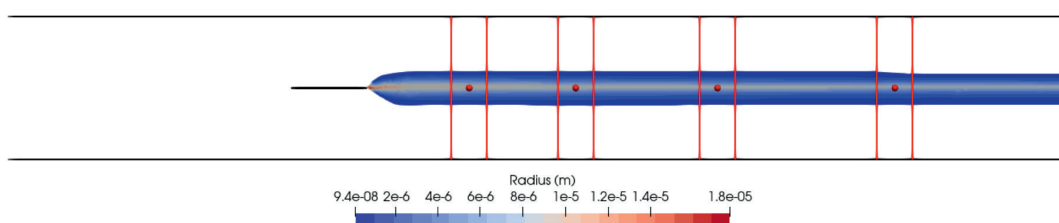


Рис. 14. Распределение размеров капель и полос, на которых локально рассчитывается SMD

Fig. 14. Distribution of droplet sizes and bands on which the SMD is calculated locally

Сравнение SMD, рассчитанного в полосках и соответствующего SMD, измеренного экспериментально, можно увидеть на рис. 15

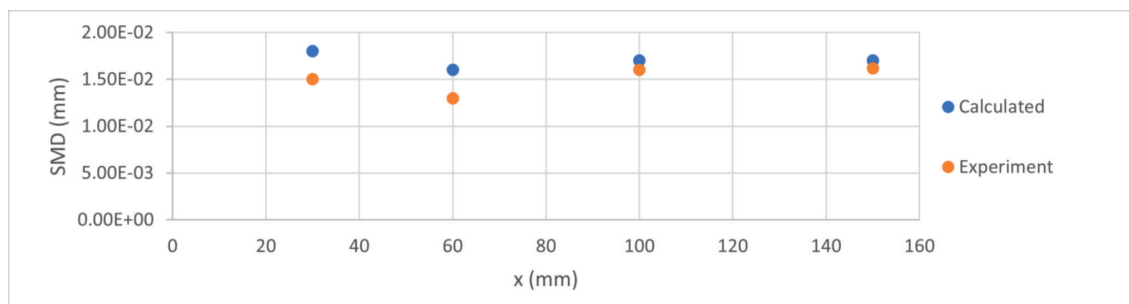


Рис. 15. Сравнение SMD между расчетными и экспериментальными значениями

Fig. 15. Comparison of SMD between calculated and experimental values

Результаты, показанные на рис. 15, показывают хорошее совпадение локальных значений SMD, полученных экспериментально и рассчитанных.

### Заключение

Исследование представляет новый метод для идентификации отдельных капель и их распределения по размерам, что является требованием для многих многофазных моделей, таких как Mixture, эйлеровой, и DPM. В отличие от традиционных методов, которые полагаются на экспериментальные данные для определения корреляций размера капель, этот метод использует распределение объемной доли модели VOF. Модель VOF, хотя и требует значительных вычислительных ресурсов, не нуждается в заранее определенных распределениях размера капель для функционирования. Метод был проверен в двух тестах:

- распад жидкой пленки: результаты показали, что на ранних стадиях рассчитанные значения SMD значительно отличаются от измеренных значений, но со временем они сходятся, что демонстрирует эффективность метода в определении размеров капель на основе сходящегося численного решения распределения объемной доли жидкости;
- распыл керосина в предварительном смесительном канале для камер сгорания газовых турбин при высоком давлении: результаты продемонстрировали хорошее совпадение локальных значений SMD, полученных экспериментально и рассчитанных по методу.

При сравнении метода локальных максимумов с первым методом (методом выделения и идентификации сгустков фрагментов жидкости), первый метод превосходит в определении недавно сформированных капель в областях, где присутствуют как капли, так и жидкостная струя. Это связано с тем, что первый метод определяет каждую каплю на основе формы и предварительных знаний о максимальных и минимальных размерах капель. Однако метод локальных максимумов не может различать струю и распавшуюся часть жидкости. В областях, где жидкость полностью распалась, метод локальных максимумов будет функционировать с хорошей точностью, без необходимости задавать какие-либо параметры, в то время как первый метод все еще требует их.

При сравнении метода локальных максимумов со вторым методом (методом пороговых значений) оба метода требуют полного распада струи жидкости. Однако второй метод требует пороговых значений для работы, в то время как метод локальных максимумов не требует таких параметров.

## Список литературы

1. Ansys, Inc. Ansys Fluent (Version 2023). Canonsburg, PA: Ansys, Inc., 2023.
2. **Аксенов А. А., Жлуктов С. В., Каширин В. С., Сазонова М. Л., Черный С. Г., Дроздова Е. А., Роде А. А.** Численное моделирование в ПК FlowVision распыла и испарения сырья в потоке газа-теплоносителя при печном производстве технического углерода // Компьютерные исследования и моделирование, 2023. Т. 15. № 4. С. 921–939. DOI: 10.20537/2076-7633-2023-15-4-921-939
3. **Aksenov A., Zhluktov S., Kashirin V., Sazonova M., Cherny S., Zeziulin I., Drozdova E., Rode A.** Numerical modeling of raw materials atomization and vaporization in a heat carrier gas flow in technical carbon production based on the Euler approach. E3S Web of Conferences, 2023, XXXIX Siberian Thermophysical Seminar 459, vol. 04019, p. 7. DOI: 10.1051/e3sconf/202345904019.
4. **Zhu S., Roekaerts D., Pozarlik A., Van Der Meer T.** Eulerian–Lagrangian RANS Model Simulations of the NIST Turbulent Methanol Spray Flame // Combustion Science and Technology. 2015. Vol. 187, no. 7. P. 1110–1138. DOI: 10.1080/00102202.2015.1019616
5. **Chen X., Ma D., Yang V., Popinet S.** High-fidelity simulations of impinging jet atomization. Atomization Spray, 2013. Vol. 23, no. 12. P. 1079–1101.
6. **Yasuda N., Yamamura K., Mori Y. H.** Impingement of liquid jets at atmospheric and elevated pressures: an observational study using paired water jets or water and methylcyclohexane jets // Proc. R. Soc. A. 2010. Vol. 466. P. 3501–3526. DOI: 10.1098/rspa.2010.0144
7. **Brandt M., Rachner M., Schmitz G.** An Experimental and Numerical Study of Kerosine Spray Evaporation in a Premix Duct for Gas Turbine Combustors at High Pressure // Combustion Science and Technology. 1998. Vol. 138, no. 1–6. P. 313–348. DOI: 10.1080/00102209808952074
8. **Brackbill J. U., Kothe D. B., Zemach C.** A continuum method for modeling surface tension // J. Comput. Phys. 1992. Vol. 100, no. 2. P. 335–354. DOI: 10.1016/0021-9991(92)90240-Y
9. **Vachaparambil K. J., Einarsrud K. E.** Comparison of Surface Tension Models for the Volume of Fluid Method // Processes. 2019. Vol. 7, no. 8. P. 542. DOI: 10.3390/pr7080542
10. **Schiller L., Naumann Z.** Ver. Deutsch. Ing., 1935. Vol. 77. P. 318.
11. **Ranz W. E., Marshall W. R. Jr.** Vaporation from Drops, Part I // Chem. Eng. Prog. 1952. Vol. 48, no. 3. P. 141–146.
12. **Ranz W. E., Marshall W. R. Jr.** Vaporation from Drops, Part I and Part II // Chem. Eng. Prog. 1952. Vol. 48, no. 4. P. 173–180.
13. **German R. M.** Powder Injection Moulding. MPIF, Princeton, New Jersey, 1990.
14. **Schmidt D. P., Corradini M. L., Rutland C. J.** A two-dimensional, non-equilibrium model of flashing nozzle flow. In Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference, 1999, San Francisco, July 18–23; ASME, New York.
15. **Сами М., Шютце Й., Хатчинсон П., Озаркар С.** Эффективное моделирование первичного распыления жидкости с помощью эйлерово-лагранжевой гибридной модели в ANSYS Fluent. 2019.
16. **Сипатов А. М., Карабасов С. А., Гомзинов Л. Ю., Абрамчук Т. В., Семаков Г. Н.** Моделирование процесса распыла с использованием адаптивных сеточных моделей // Вычислительная механика сплошных сред. 2015. Т. 8, № 1. С. 93–101. DOI: 10.7242/1999-6691/2015.8.1.8
17. **Sun Y., Li Y., Drebler L., Nishad K., Sadiki A.** Multiscale numerical modeling of a complete spray evolution including breakup of liquid jet injection in gaseous cross flow // International Journal of Multiphase Flow. 2024. Vol. 170. P. 104655. DOI: 10.1016/j.ijmultiphaseflow.2023.104655
18. **Spitzenberger A., Neumann S., Heinrich M., Schwarze R.** Particle detection in VOF simulations with OpenFOAM // SoftwareX. 2020. Vol. 11. P. 100382.

19. **Wu P. K., Kirkendall K. A., Fuller R. P., Nejad A. S.** Spray structures of liquid jets atomized in subsonic crossflows // *J. Propul. Power*. 1998. Vol. 14, no. 2. P. 173–182. DOI: 10.2514/2.5283
20. **Cherny S. G., Zeziulin I. V., Tarraf D., Pochuev A. P., Boronenko I. S.** The direct and inverse problems of modeling the carbon black furnace production process // *Eurasian Journal of Mathematical and Computer Applications*, 2025.

## References

1. Ansys, Inc. Ansys Fluent (Version 2023). Canonsburg, PA: Ansys, Inc., 2023.
2. **Aksenov A. A., Zhlukto S. V., Kashirin V. S., Sazonova M. L., Cherny S. G., Drozdova E. A., Rode A. A.** Numerical modeling in the FlowVision PC of spraying and evaporation of raw materials in the flow of coolant gas during furnace production of carbon black. *Computer Research and Modeling*, 2023, vol. 15, no. 4, pp. 921–939. (in Russ.) DOI: 10.20537/2076-7633-2023-15-4-921-939
3. **Aksenov A., Zhlukto S., Kashirin V., Sazonova M., Cherny S., Zeziulin I., Drozdova E., Rode A.** Numerical modeling of raw materials atomization and vaporization in a heat carrier gas flow in technical carbon production based on the Euler approach. *E3S Web of Conferences*, 2023, XXXIX Siberian Thermophysical Seminar 459, vol. 04019, p. 7. DOI: 10.1051/e3sconf/202345904019
4. **Zhu S., Roekaerts D., Pozarlik A., Van Der Meer T.** Eulerian–Lagrangian RANS Model Simulations of the NIST Turbulent Methanol Spray Flame. *Combustion Science and Technology*, 2015, vol. 187, no. 7, pp. 1110–1138. DOI: 10.1080/00102202.2015.1019616
5. **Chen X., Ma D., Yang V., Popinet S.** High-fidelity simulations of impinging jet atomization. *Atomization Spray*, 2013, vol. 23, no. 12, pp. 1079–1101.
6. **Yasuda N., Yamamura K., Mori Y. H.** Impingement of liquid jets at atmospheric and elevated pressures: an observational study using paired water jets or water and methylcyclohexane jets. *Proc. R. Soc. A*, 2010, vol. 466, pp. 3501–3526. DOI: 10.1098/rspa.2010.0144
7. **Brandt M., Rachner M., Schmitz G.** An Experimental and Numerical Study of Kerosine Spray Evaporation in a Premix Duct for Gas Turbine Combustors at High Pressure. *Combustion Science and Technology*, 1998, vol. 138, no. 1–6, p. 313–348. DOI: 10.1080/00102209808952074
8. **Brackbill J. U., Kothe D. B., Zemach C.** A continuum method for modeling surface tension. *J. Comput. Phys.*, 1992, vol. 100, no. 2, pp. 335–354. DOI: 10.1016/0021-9991(92)90240-Y
9. **Vachaparambil K. J., Einarsrud K. E.** Comparison of Surface Tension Models for the Volume of Fluid Method. *Processes*, 2019, vol. 7, no. 8, p. 542. DOI: 10.3390/pr7080542
10. **Schiller L., Naumann Z.** *Ver. Deutsch. Ing.*, 1935, vol. 77, p. 318.
11. **Ranz W. E., Marshall W. R. Jr.** Vaporization from Drops, Part I. *Chem. Eng. Prog.*, 1952, vol. 48, no. 3, p. 141–146.
12. **Ranz W. E., Marshall W. R. Jr.** Vaporization from Drops, Part I and Part II. *Chem. Eng. Prog.*, 1952, vol. 48, no. 4, p. 173–180.
13. **German R. M.** Powder Injection Moulding. MPIF, Princeton, New Jersey, 1990.
14. **Schmidt D. P., Corradini M. L., Rutland C. J.** A two-dimensional, non-equilibrium model of flashing nozzle flow. In *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, 1999, San Francisco, July 18–23; ASME, New York.
15. **Sami M., Schuetze J., Hutchinson P., Ozarkar S.** Efficiently Modeling Primary Liquid Atomization using an Eulerian Lagrangian Hybrid Model in ANSYS Fluent. 2019. (in Russ.)
16. **Sipatov A. M., Karabasov S. A., Gomzikov L. Yu., Abramchuk T. V., Semakov G. N.** Atomization modelling using adaptive mesh refinement modeling liquid jet atomization using adaptive mesh refinement. *Computational Continuum Mechanics*, 2015, vol. 8, no. 1, pp. 93–101. (in Russ.) DOI: 10.7242/1999-6691/2015.8.1.8

17. **Sun Y., Li Y., Drebler L., Nishad K., Sadiki A.** Multiscale numerical modeling of a complete spray evolution including breakup of liquid jet injection in gaseous cross flow. *International Journal of Multiphase Flow*, 2024, vol. 170, p. 104655. DOI: 10.1016/j.ijmultiphaseflow.2023.104655
18. **Spitzenberger A., Neumann S., Heinrich M., Schwarze R.** Particle detection in VOF simulations with OpenFOAM. *SoftwareX*, 2020, vol. 11, p. 100382.
19. **Wu P. K., Kirkendall K. A., Fuller R. P., Nejad A. S.** Spray structures of liquid jets atomized in subsonic crossflows. *J. Propul. Power*, 1998, vol. 14, no. 2, pp. 173–182. DOI: 10.2514/2.5283
20. **Cherny S. G., Zeziulin I. V., Tarraf D., Pochuev A. P., Boronenko I. S.** The direct and inverse problems of modeling the carbon black furnace production process. *Eurasian Journal of Mathematical and Computer Applications*, 2025.

### Информация об авторах

**Чёрный Сергей Григорьевич**, доктор физико-математических наук, профессор  
ResearcherID: L-7062-2017

**Тарраф Даниэль**, аспирант

### Information about the Authors

**Sergey G. Cherny**, Doctor in Physics and Mathematics, Professor  
ResearcherID: L-7062-2017

**Tarraf Daniel**, PhD Student

*Статья поступила в редакцию 17.02.2025;  
одобрена после рецензирования 22.02.2025; принята к публикации 22.02.2025*  
*The article was submitted 17.02.2025;  
approved after reviewing 22.02.2025; accepted for publication 22.02.2025*

### Правила оформления текста рукописи

Авторы представляют статьи на русском или английском языке объемом от 0,5 авторского листа (20 тыс. знаков) до 1 авторского листа (40 тыс. знаков), включая иллюстрации (1 иллюстрация форматом  $190 \times 270$  мм = 1/6 авторского листа, или 6,7 тыс. знаков). Публикации, превышающие указанный объем, допускаются к рассмотрению только после индивидуального согласования с редакцией журнала.

Текст рукописи должен быть представлен в редколлегию в виде файла MS Word (.doc, .docx). Гарнитура Times New Roman, размер шрифта 11, межстрочный интервал 1, размеры полей – стандартные значения текстового редактора. Форматирование – выравнивание по ширине страницы, переносы слов включены, каждый новый абзац начинается с красной строки. Не допускается ручное форматирование абзацев (пробелами, лишними переводами строк, разрывами страниц).

### Структура статьи

- Индекс УДК (универсальной десятичной классификации). Выравнивание по левому краю
- Название статьи. Выравнивание по центру, полужирный шрифт
- ФИО авторов (полностью). Выравнивание по центру, полужирный шрифт
- Места работы всех авторов. Выравнивание по центру, курсив
- Адреса электронной почты, ORCID авторов
- Аннотация статьи
- Ключевые слова, не более 10
- Благодарности, сведения о финансовой поддержке
- Название статьи **на английском языке**. Выравнивание по центру, полужирный шрифт
- ФИО авторов **на английском языке** (полностью). Выравнивание по центру, полужирный шрифт
- Места работы авторов **на английском языке**. Выравнивание по центру, курсив
- Аннотация статьи **на английском языке (Abstract)**, 200–250 слов
- Ключевые слова **на английском языке (Keywords)**, не более 10
- Благодарности, сведения о финансовой поддержке **на английском языке**, если есть соответствующий раздел на русском языке (**Acknowledgements**)
- Основной текст
- Список литературы / **References**
- Сведения об авторах

### Требования к оформлению основного текста и иллюстративных материалов

Основной текст должен быть представлен в структурированном виде, рекомендуется использовать подзаголовки – например: Введение, Методика..., Выводы, Результаты, Заключение.

Подзаголовки отделяются и набираются полужирным шрифтом. В целях выделения частей текста и отдельных слов и словосочетаний допускается использование курсива или полужирного шрифта. Подчеркивание, разрядка, изменение основного кегля и выделение цветом не используются.

Иллюстрации к рукописи статьи должны быть приложены в виде отдельных файлов. При этом в тексте должно содержаться включенное изображение с указанием имени файла. Все иллюстрации, содержащие схемы, графики, алгоритмы и т. п., должны быть представлены в векторном виде (.ai, .eps, .cdr). Скриншоты и другие растровые изображения должны быть представлены в максимально высоком качестве, без каких-либо потерь и искажений (.jpg, .tif). Все иллюстрации должны иметь подрисуночную подпись – свое название. Надписи к таблицам и подписи к иллюстрациям приводятся **на двух языках (русском и английском)**.

Примеры:

*Рис. 1. Диаграмма производительности...*

*Fig. 1. Performance diagram...*

*Таблица 1*

*Сравнение алгоритмов...*

*Table 1*

*Comparison of algorithms...*

Нумерация последовательная и неразрывная от начала статьи. Не допускается использование других наименований, кроме «Рис.» / «Fig.», «Таблица» / «Table», и усложнение нумерации (например, «Рис. 3.2.»). Ссылка на иллюстрацию в тексте должна быть приведена в круглых скобках, например: (рис. 1), (табл. 1).

Формулы должны быть набраны с использованием редактора MathType либо встроенного редактора формул MS Word. Кегль основных символов – 11, греческие символы набираются прямым шрифтом, латинские – курсивом. Нумеруются только те формулы, на которые автор ссылается в тексте.

## Abstract

Аннотация статьи на английском языке (Abstract) не должна быть дословным переводом русскоязычной аннотации. Раздел Abstract, как и основной текст, должен быть структурирован, в нем должно содержаться описание цели работы, методов исследования, научной значимости, выводов / результатов. Требуется качественный перевод на английский язык (при необходимости просим авторов обращаться к профессиональным переводчикам). **Объем Abstract 200–250 слов.**

## Список литературы / References

Список литературы и список литературы на английском языке (References) размещаются в общем разделе. Рекомендуемое количество цитируемых в статье источников – не менее 10, в список желательно включать ссылки на актуальные работы по теме исследования, особенно в иностранных периодических изданиях.

В тексте статьи ссылки на литературу указываются цифрами в квадратных скобках, при необходимости указываются номера страниц, например: [2; 3. С. 15].

Список литературы нумеруется в порядке цитирования и оформляется в соответствии с ГОСТ Р 7.0.5-2008 на библиографическое описание (знаки тире в описании опускаются). Ссылки на неопубликованные работы, а также на интернет-ресурсы (кроме электронных изданий, поддающихся библиографическому описанию) оформляются в виде сноски.

В Список литературы ссылки на источники следует включать на оригинальном языке опубликования. Каждый источник должен быть также оформлен на английском языке (References) по международному стандарту для публикаций в области информатики IEEE Style со следующими отличиями:

- инициалы авторов указываются после фамилии;
- название статьи не берется в кавычки, отделяется точкой;

- отсутствует союз «and» перед фамилией последнего автора;
- в диапазоне страниц – удвоенная «р» (например, «рр. 2–9»);
- год издания указывается после места издания (для книг) и сразу после названия журнала (для периодики).
- Перевод источника на английский язык:
- если источник имеет выходные данные на английском языке, то для формирования References **следует использовать именно эти данные**;
- если оригинальная публикация не содержит выходных данных на английском языке, то допускается транслитерация названия материала на латинский алфавит в сочетании с переводом на английский язык в квадратных скобках. В конце описания указывается, на каком языке написана эта работа, например, (in Russ.). При транслитерации можно воспользоваться интернет-ресурсом <http://ru.translit.ru/>, рекомендуется выбрать стандарт BSI. Место издания не транслитерируется, указывается полностью на английском языке, например: Moscow. Название издательства / издателя, как правило, транслитерируется. Для журналов, у которых есть официальное название на английском языке, – использовать его (проверить на сайте журнала, или, например, в библиотеке WorldCat), если названия на английском языке нет, использовать транслитерацию по системе BSI. Не следует самостоятельно переводить названия журналов.

Если у цитируемого источника есть **цифровой идентификатор DOI** (<https://search.crossref.org/>), его требуется обязательно указывать в конце библиографической ссылки.

Примеры оформления ссылок. Каждый источник в том же пункте дублируется на английском языке (References).

***Источник на русском языке, перевод на английский доступен в метаданных статьи***

1. Журавлев С. С., Рудометов С. В., Окольников В. В., Шакиров С. Р. Применение модельно-ориентированного проектирования к созданию АСУ ТП опасных промышленных объектов // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 56–67. DOI 10.25205/1818-7900-2018-16-4-56-67

Zhuravlev S. S., Rudometov S. V., Okolnishnikov V. V., Shakirov S. R. Model-Based Design Approach for Development Process Control Systems of Hazardous Industrial Facilities. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, pp. 56–67. (in Russ.) DOI 10.25205/1818-7900-2018-16-4-56-67

***Источник на английском языке. Оформляем согласно требованиям для References. Приводим только 1 раз.***

2. Telnov V. I. Optimization of the Beam Crossing Angle at the ILC for E + e- and yy Collisions. *Journal of Instrumentation*, 2018, vol. 13, no. 03, pp. P03020–P03020. DOI 10.1088/1748-0221/13/03/p03020

***Метаданные источника доступны только на русском языке***

3. Жижимов О. Л., Федотов А. М., Шокин Ю. И. Технологическая платформа массовой интеграции гетерогенных данных // Вестник НГУ. Серия: Информационные технологии. 2013. Т. 11, вып. 1. С. 24–41.

Zhizhimov O. L., Fedotov A. M., Shokin Yu. I. Tekhnologicheskaya platforma massovoi integratsii geterogennykh dannykh [Technology Platform for the Mass Integration of Heterogeneous Data]. *Vestnik NSU. Series: Information Technologies*, 2013, vol. 11, no. 1, pp. 24–41. (in Russ.)

## Сведения об авторах

Последний раздел статьи – информация об авторе / авторах **на русском и английском языках**:

- ФИО полностью, ученая степень, ученое звание;
- идентификаторы автора, такие как ResearcherID (всем авторам рекомендуется использовать данные сервисы для ведения актуального списка своих публикаций);
- контактный телефон (не публикуется).

Если статья представляется на английском языке, необходимо приложить перевод на русский язык названия, аннотации, ключевых слов, сведений об авторе.

### **Доставка материалов**

Материалы предоставляются в редакцию по электронной почте [inftech@vestnik.nsu.ru](mailto:inftech@vestnik.nsu.ru).

### **Порядок рецензирования**

Все статьи сначала проходят проверку на заимствование и только после этого отправляются на рецензирование. Редакционный совет не допускает к публикации материал, если имеется достаточно оснований полагать, что он является плагиатом.

Тип рецензирования статей – двухуровневое, одностороннее анонимное («слепое»).

Для каждой статьи редколлегией выбираются рецензенты, научная деятельность которых связана с темой представленного материала. Ответственный секретарь журнала обращается к ним с просьбой дать экспертную оценку статье либо помочь организовать рецензирование.

Рецензии для журнала «Вестник НГУ. Серия: Информационные технологии» составляются по единой схеме и подразумевают оценку по следующим критериям: соответствие тематике журнала, оригинальность и значимость результатов, качество изложения материала.

Заполненный бланк рецензии высылается на электронный адрес редакции. В зависимости от экспертных заключений статья может быть принята редакционным советом к опубликованию, рекомендована автору к доработке (с последующим повторным рецензированием либо без него) или отклонена (с предоставлением автору мотивированного отказа). Автору на электронный адрес высылается текст рецензии без указания ФИО рецензента и его контактных данных.

Все рецензии хранятся в редакции журнала не менее 5 лет. Редколлегия журнала обязуется при поступлении соответствующего запроса направлять копии рецензий в Министерство науки и высшего образования Российской Федерации.